The Thesis Committee for Yin-Jui Chang
certifies that this is the approved version of the following thesis:

Signal Translation between EEG and ECoG to improve non-invasive based BCI performance

Committee:

Michael D. Bryant, Supervisor

Benito R. Fernandez, Co-Supervisor

Robert J. Buchanan

Signal Translation between EEG and ECoG to improve non-invasive based BCI performance

by

Yin-Jui Chang

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2018

Acknowledgments

I would like to give special thanks to Professor Michael Bryant who provided the opportunity and encouragement for this research. I would also like to thank him for offering financial support to the author by appointing him as his teaching assistant. His guidance helped me throughout the research and thesis writing period.

In addition, I am sincerely grateful to Professor Benito Fernandez and Professor Robert Buchanan for introducing me to the key concepts of the applied intelligence and Brain-Computer Interface. I am really thankful to them for contributing a significant amount of time to proof read and edit this thesis, and their guidance helping me scramble over the obstacles over the research as well.

Eventually, I also appreciate the help from Juan Sebastian Rincon, Gautam Krishna, Gaudi Karina Morantes Quintana and many other students from the University of Texas at Austin.

Yin-Jui Chang

The University of Texas at Austin

May 2018

Signal Translation between EEG and ECoG to improve non-invasive based BCI performance

Yin-Jui Chang, M.S.E.
The University of Texas at Austin, 2018

Supervisor: Michael D. Bryant, Benito R. Fernandez

An electroencephalography (EEG) / electrocorticography (ECoG) inverse model for the Brain-Computer Interface (BCI) was developed, and the analysis of the signals was simulated in Python environment. The inverse solution, in an attempt to estimate ECoG from EEG, can significantly improve the performance of non-invasive based BCI. NonLinear Principal Component Analysis (NLPCA) is employed to reduce the complexity of computation. Forward model is then derived from the electro-physiological perspective to capture the dynamic of the signals. To represent nonlinear approximations, a NeuroBondGraph (NBG) approach is introduced to model both the system dynamics and the nonlinearity in a more efficient way. Inverse solution is then established integrating with the de-mapping part of NLPCA. The simulation results are demonstrated by the comparison between original signals and reconstructed signals from our model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

In recent years, Brain-Computer Interface (BCI), or so-called "Brain-Machine Interface" (BMI) has become a popular research topic [1]. It is a system that utilizes the user's thought to interact with external devices. Three-dimensional control of a robotic arm for assistive devices and rehabilitation engineering has been demonstrated possible by a patient with tetraplegia using signals from motor cortex to control a robotic arm to perform reach and grasp movements [2]. Environmental control integrated with BCI and a robotic platform is a new technology that allows the user the ability to control the television, lights, a thermostat, or any domestic device. In addition, a promising application for BCI was recently demonstrated by a spinal-cord-injured patient, who was able to kick off the ball for the 2014 FIFA's World Cup [3] by using a brain-controlled prosthetic device. Another encouraging example is Battelle NeuroLife system. It assisted Ian Burkhart, a quadriplegic due to a diving accident, to recover the ability to perform some daily tasks, including opening and closing his hand, grasping and swiping the credit card, picking up and holding a phone to his ear, and even playing a guitar video game with his own fin-

gers and hands. Those daily movements that seemed only impossible for paralyzed patients can now be performed with BCI.

In addition, BCI can be formulated to transfer and manipulate information bi-directionally, either altering the input of information to the brain or altering the output of information from the brain. Take cochlear implants and functional retinal prosthesis for example [4]. Environmental stimuli such as light and sound can be captured and translated into the codes to be transmitted back into the human brain. Moreover, sensory feedback such as proprioception, touch, and pain from the body allows the subject to continuously adjust motor commands to the body. Obviously, once BCI can be fully implemented, patients with neurological disorder would have improved quality of life, and even normal individuals may be able to augment the level of performance.

Therefore, the key part in the BCI system is how well we can interpret the user's intention. If we cannot have an accurate interpretation, we cannot have an accurate corresponding interface. Brain signals acquisition can be categorized into two methods: invasive method and noninvasive method. The main difference between these two methods is the need for brain surgery.

The noninvasive methods do not require brain surgery, so it is a safe and convenient approach to obtain brain information. Electroencephalography (EEG) and Magnetoencephalography (MEG) are two commonly used noninvasive measurements. EEG is used for detecting the electrical activity in the human brain by placing the recording electrodes on the scalp [5]. However, due to the distortion effects of the tissues between the signal source and the recording electrodes, it is nearly impossible to detect the firing pattern of individual neurons. Rather than electrical activity,

MEG is used to detect tiny magnetic fields outside the skull generated by electrical currents in neurons by a superconducting quantum interference device (SQUID) [6]. It provides higher spatial and temporal resolution, but it is not yet field-deployable and requires expensive technologies and a highly-shielded chamber.

The invasive methods are a way to improve the resolution and quality of recordings. Electrocorticography (ECoG), also known as intracranial electroencephalography (iEEG), is one of the most commonly employed approaches. It uses stainless steel or platinum electrodes to capture brain activity directly from the surface of the cerebral cortex, where it eliminates the distortion effects from the skull and intermediate tissue, hence, providing better ability to map important functional areas of the brain. To obtain even higher signal fidelity, multi-electrode arrays (MEAs), also called microelectrode arrays, have been developed for local field potential (LFP) detection [7]. LFP is the aggregated signal of electrical potentials from a larger population of neurons near the recording site. Despite high signal quality, it is limited to a relatively small local region of the brain underneath the electrode array.

Nowadays, EEG source localization is a hot topic [8, 9, 17] and has several potential applications. The technique is to find the brain areas or even individual neurons responsible for EEG information of interest. To achieve the goal, forward and inverse problems are necessary to be dealt with. The forward problem is to estimate the potentials at the EEG electrodes with a given electrical source. The information and the evaluation obtained from forward problem play a significant role in solving the inverse problem. Literally, the so-called inverse problem is to inverse the forward problem mentioned above: finding the electrical sources which are responsible for the EEG signals measured at the electrodes. Such a technique plays a significant role in improving the performance of noninvasive based BCI.

1.2 Review on EEG source localization solutions

1.2.1 Forward solution

We follow [17] presentation on this topic. Generally, every neuron is regarded as a current dipole due to its communication mechanism. An action potential induces a neuron to release a chemical neurotransmitter at a synapse, either exciting or inhibiting another neuron from firing its own action potential [10]. When neurotransmitters are released at a synapse, they will cause ion channels to open and then an influx of positive ions depolarize the neuron. The local extracellular environment would set up a voltage difference along the axis of the neuron due to depletion of positive ions. Accordingly, every neuron can be thought of as an electrical current dipole with a specific orientation and polarity.

The first model describing the head is a single homogeneously conducting compartment.

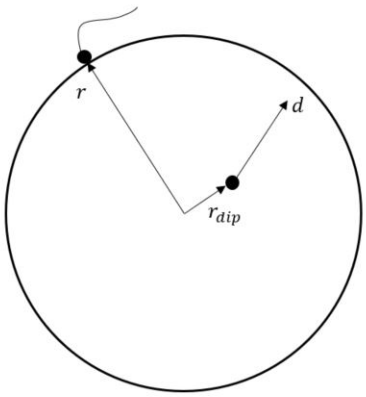
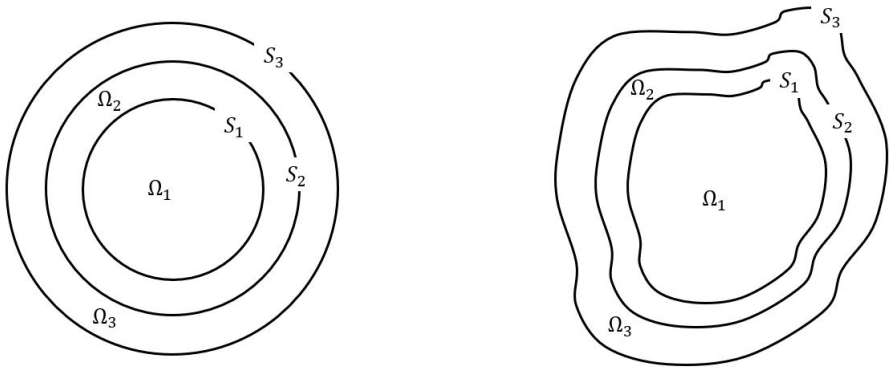


Figure 1.1: Single conducting compartment model. Modified from [9].

The electrical potential is derived as follow:

$$V(\ \vec{r}\ ,\vec{r}_{dip}\ ,\vec{d}\ ) = \frac{\vec{d}\,(\vec{r}-\vec{r}_{dip})}{4\pi\sigma\|\vec{r}-\vec{r}_{dip}\|^3} \tag{1.1}$$

where σ represents the conductivity, $\vec{d}$ represents the dipole moment, $\vec{r}$ and $\vec{r}_{dip}$ are the position where the potential is measured and the position where the dipole is. Then a three layer model with the shape as either a sphere or a realistic head is introduced to model the head due to the different conductivities between layers: brain, skull, and scalp. Each layer is assumed to be an infinite homogeneous isotropic conductor.



(a) Three concentric sphere model (b) Three layer realistic head model

Figure 1.2: Three layer models. Modified from [13].

The capacitive component of tissue impedance, and the electromagnetic wave effect are ignored to simplify the model, while maintaining a good approximation to the current density, J, with the conductivity, σ, electrical field, E, and electrical potential, V:

$$J = \sigma E = -\sigma \nabla V \tag{1.2}$$

where ∇ represents the gradient operator.

Due to these properties of body tissues, the currents depend on the sources and also obey the principle of superposition. Here we represent the sources by a contribution of impressed current density or a current dipole moment per unit volume, J_i. Then, the above equation is modified as follows:

$$J = \sigma E + J_i = -\sigma \nabla V + J_i \tag{1.3}$$

According to Ampere's circuit law in Maxwell's equations,

$$\nabla \times \frac{B}{\mu} = J \tag{1.4}$$

After performing the divergence of the above equation, we can get the following:

$$\nabla \cdot (\nabla \times \frac{B}{\mu}) = \nabla \cdot J = \nabla \cdot (-\sigma \nabla V + J_i) = 0 \tag{1.5}$$

Then,

$$\nabla \cdot \sigma \nabla V = \nabla \cdot J_i \tag{1.6}$$

Letting x_0 and q be the dipole position and moment respectively, and considering the homogeneous Neumann boundary conditions, the problem becomes the following:

$$\begin{aligned} \sigma(x)\nabla^2 V(x) &= \nabla \cdot (q\delta(x - x_0)), \quad x \in \Omega_i \\ V(x^+) &= V(x^-), \\ \sigma(x^-)\nabla V(x^-) \cdot n &= \sigma(x^+)\nabla V(x^+) \cdot n, \quad x \in S_i \end{aligned} \tag{1.7}$$

where, x is any point in the domain $\Omega = \{\, \Omega_1, \Omega_2, \Omega_3 \,\}$ or on the surface $S = \{\, S_1, S_2, S_3 \,\}$, and ∇^2 is the Laplacian operator. For any point $x \in S_i$, we define $x^- \in \Omega_i$ is the limit point inside the surface S_i, and $x^+ \in \Omega_{i+1}$ is the limit point

outside the surface S_i.

To solve the above equations, the boundary element approach is utilized with the assumption that the conductivity is isotropic and constant. The *boundary element method* (BEM) is a numerical technique that generates meshes to construct the interfaces between layers. It has been demonstrated with high accuracy and efficiency by employing appropriate integral formulations [11].

However, in reality, BEM cannot be applied since there exists inhomogeneities and the medium has an anisotropic conductivity. For such a case, the *finite element method* (FEM) was used to construct a three dimensional model for the head [12], allowing to consider accurate inhomogeneous approximation, increasing the computational complexity. Instead of generating the mesh, a numerical solution called *finite points mixed method* (FPMM) that only uses nodes to be able to solve the equations above [13]. Such a method significantly reduces the complexity of computation.

1.2.2 Inverse solution

The EEG inverse problem is an ill-posed problem. Given a mapping equation $y = K(x)$, only noisy data y^d can be observed, and thus it is very hard to estimate noiseless x. Moreover, the dimension of x (a hundred billion of neurons) is much higher than that of y (Up to 300 EEG channel can be placed on the scalp). The mapping function is not bijective, hence, the solution is non-unique and unstable. In the literature, various mathematical models introduce constraints on either the dipole positions, magnitudes, or orientations, in an attempt to remedy such situation.

We follow [9] review on this topic, there are two approaches to solve the inverse problem: non-parametric and parametric methods. The main difference between these two approaches is whether the number of dipoles is assumed *a priori* or not. Non-parametric method is to estimate the dipole magnitude and orientation of dipoles at fixed positions while parametric method is to estimate the dipole parameters of an *a priori* number of dipoles.

Non-parametric method

Regularization is the most commonly used approximation of an ill-posed problem by discovering the best-approximate solution in the case that only the difference between noisy data y^δ and the noiseless data y is less than δ. This approach is to find x_α^δ that minimizes the following function:

$$F_\alpha(x) = \|Kx - y^\delta\|^2 + \alpha L(x) \tag{1.8}$$

where the regularization parameter α is determined by the balance between the perturbation error in y and the regularization error in the regularized solution. This can be done by several methods, including L-curve method, general-cross validation method, composite residual and smoothing operator, minimal product method, and zero crossing [9].

A number of researches have shown the solution with different regularization terms $L(x)$. Minimum norm estimates (MNE) are to find the solution with minimum power according to Tikhonov regularization $L(x) = \|x\|^2$. Weighted minimum norm estimates (WMNE) is another algorithm introducing a weighting matrix W in the term, where $L(x) = \|Wx\|^2$. Grech et al. state that when using minimum norm methods, deeper sources could not be recovered since the dipoles located at the surface of the source space with smaller magnitude would be obtained with higher chance for the

same dataset.

Low resolution electrical tomography (LORETA) [14] combines the lead-field normalization with the Laplacian operator, providing the same opportunity of being reconstructed for all the sources, including the ones close to the surface and the deeper ones. This method selects the regularization term as follow:

$$L(x) = \|BWx\|^2 \tag{1.9}$$

where, B is the discrete 3D Laplacian operator, and W here is a diagonal matrix representing the normalization of the lead field matrix K.

Other methods that stem from MNE and WMNE are standardized low resolution brain electromagnetic tomography (sLORETA) and variable resolution electrical tomography (VARETA). The former one uses the estimation given by MNE and standardizes it by using its variance; while the latter one is the WMNE solution where the regularization parameter varies spatially at each point.

Quadratic regularization (QR) is another method that uses the gradients of the dipole intensity, where $L(x) = \|\nabla x\|^2$. Spatial regularization is the modification of QR, providing the ability to detect the intensity jumps. When considering the time effect, a spatio-temporal regularization (ST-MAP) assumes that x_{t-1} and x_t may be very close to each other. Such a method has demonstrated that a better solutions can be obtained by using Kalman filter with appropriate dynamic models [9].

Instead of using regularization method, the Backus-Gilbert method is to approximate the inverse operator of matrix K, where the EEG data y will be projected onto the solution space.

In norm minimization approach mentioned above, several assumptions are made to obtain the optimal mathematical solution, but those assumptions are purely mathematical without incorporating biophysical constraints. Local AUtoRegressive Average (LAURA) introduces the biophysical rules based on the Maxwell's laws of electromagnetic field into the minimum norm solution.

Parametric method

Parametric methods are known as Equivalent Current Dipole Methods or Spatio-Temporal Dipole Fit Models. Searching for the dipole positions and orientations is necessary in such an approach. Beamforming methods employ a spatial filter that only the signals from the sources of interest will be obtained, allowing us to estimate the neural activity at any location without the prior knowledge of the number of sources and then the anatomical information can be easily included [9].

Brain electric source analysis (BESA) is a dipole-fit model finding the minimization of a cost function consisting of four criteria: (1) Residual Variance: the amount of the signals unexplained by current model, (2) Source Activation Criterion: the tendency for the source to be active outside the *a priori* time interval of activation, (3) Energy Criterion: this criteria avoids the interaction between two sources where compensation happens, (4) Separation Criterion: the solution will be derived based on which as few sources as possible are active in the meantime [9].

Subspace techniques take the signal noise into consideration during dipole localization, so it provides more robust solution. Multiple-signal classification (MUSIC) algorithm first estimates the signal subspace from the data, then scans a single dipole model and eventually computes the projections onto the subspace [9, 15]. The ad-

vantage is that rather than searching for all the sources simultaneously, each source is obtained one after another. Instead of using the entire estimated subspace, first principal vector (FINES) only employs the projections onto the subspace spanned by FINES vector set, a small set of vectors. It has been shown that FINES solution has more identifiable localization results than MUSIC [16].

Alternative methods to solve the inverse source localization problem are computational intelligence algorithms, including artificial neural network and genetic algorithms. They both develop a minimization approach to obtain the optimal solutions.

1.3 Methodology

In order to improve noninvasive based BCI performance, here I study the signal translation between EEG and ECoG since ECoG-based BCI has been demonstrated in the real world. I hypothesize that estimating ECoG signals based on EEG measurements and then performing BCI system can significantly improve the performance. In this thesis, I propose a novel way to estimate ECoG signals from EEG recordings in Figure 1.3.

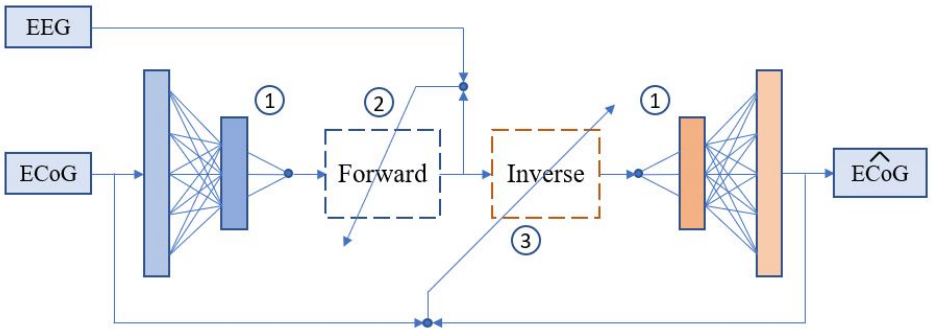


Figure 1.3: Schematic of inverse solution

First, I performed dimensionality reduction in order to reduce the computational

cost in Chapter 2. In Chapter 3, the forward model solution was derived using the NeuroBondGraph technique [26] and an inverse solution was obtained based on the forward model structure and then combining with the de-mapping part from the dimensionality reduction step. Future work is discussed in Chapter 4.

Chapter 2

Dimension Reduction for Signal Translation Model

Dimension reduction is a process to convert a set of data with high dimensions into data with less dimensions, ensuring that it still convey similar information concisely. It has several benefits:

1. It can reduce the data storage required.
2. The computational cost is decreased and the computational time will be faster.
3. It is helpful to remove the redundant features and the noise of the data, so that the performance of the model can be improved.

There are several methods able to perform dimension reduction nowadays, such as backward feature elimination [18], and factor analysis [19]. However, the most common technique is principal component analysis (PCA).

2.1 Principal Component Analysis

Principal component analysis is used to find a new set of dimensions such that all the dimensions are linearly independent [20]. Those new dimensions will be ranked based on the variance of the data along them. Higher variance means that the spread of the data is larger, and thus such a dimension is more important.

The first step to do PCA is to calculate the covariance matrix X of the data points. Secondly, we calculate the eigenvectors and the corresponding eigenvalues. The next step is to sort the eigenvectors according to their eigenvalues in a decreasing order. Then we choose the first m eigenvectors based on the summation of the normalized eigenvalues and those will be the new m dimensions of the data. Finally, we can transform the original n-dimensional data points into m dimensions. A simple example of PCA is shown in Fig. 2.1.

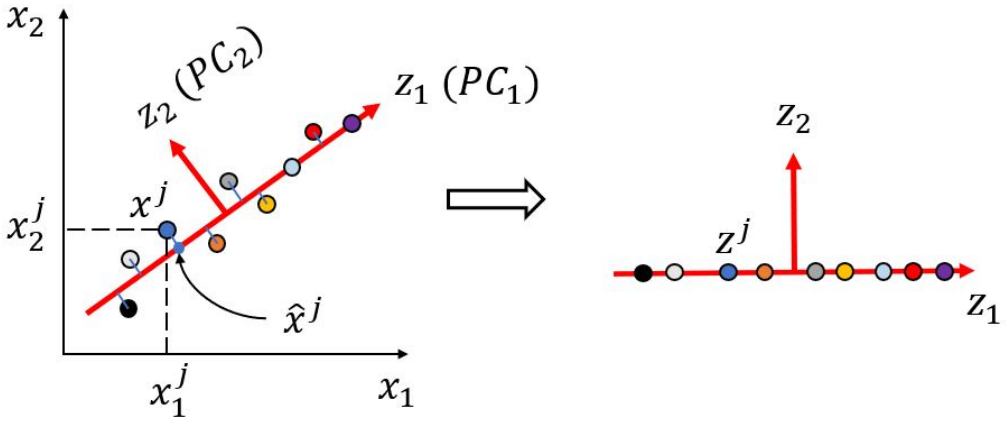


Figure 2.1: Schematic process of principal component analysis. The PCA algorithm finds eigenvectors z_1, z_2 that maximize the spread along their axis minimizing reconstruction error.

Given a dataset with the dimensions x_1 and x_2. After principal component analysis,

we can find out a dimension, z_1, that can explain the data easier, so z_1 is so-called principal component. The conversion from 2D to 1D will be helpful for further analysis or computation. If needed, the original space can be reconstructed, $\hat{x}$, in Fig. 2.1, which is the projection of x into z. PCA minimizes the error of the recovery signals. The sum of the normalized eignevalues is a measure of how much of the original data can be explained by the principal components z.

2.2 Nonlinear Principal Component Analysis (NLPCA) on ECoG Signals

The NLPCA is when the principal components of the data is a curve or a curved surface rather than a straight line or a flat surface shown in Fig. 2.2. The nonlinearity is introduced to generate better mappings. In the thesis, NLPCA is based on auto-associative neural networks (autoencoder), a specific model for artificial neural network (ANN).

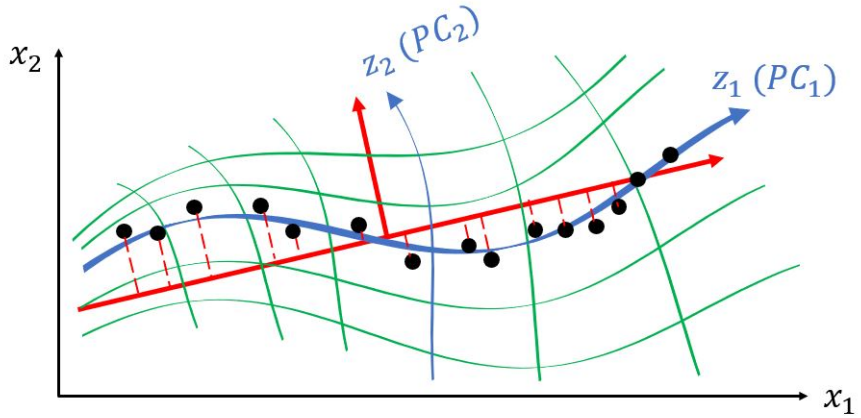


Figure 2.2: Schematic of nonlinear principal component analysis. The non-euclidean geometry of the principal components explains more information of the original data than those with euclidean geometry.

2.2.1 Mathematical model for artificial neural network

An artificial neural network (ANN) system is a computational method consisting of several simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs [22]. Such a computing system is inspired by biological neural networks in the animal's brain in Figure 2.3.

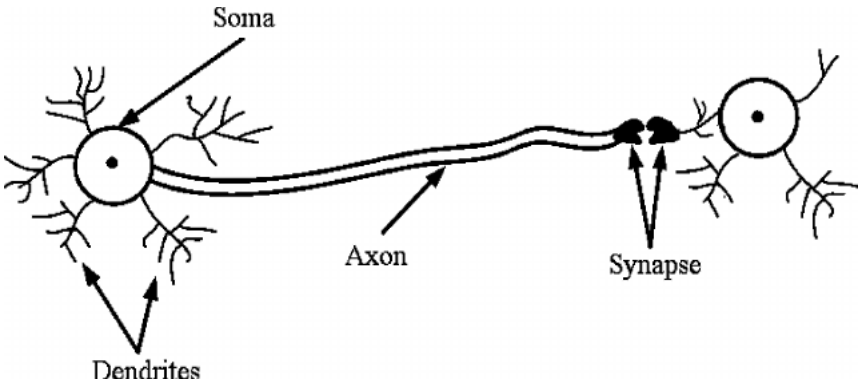


Figure 2.3: Schematic of biological neural network [23]

The **Dendrite** receives the signals from other neurons; the **Soma (cell body)** spatio-temporally aggregates all incoming signals to generate the input to the neuron; when the input reaches a threshold, the neuron fires and the signal can travel through **Axon** to other neurons; **Synapse** is the interconnection cleft of one neuron to other neurons, and the amount of signals passing through depends on the strength of the connection, so-called synaptic weight. From this biological system, we can have a mathematical expression for such a method.

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{2.1}$$

where, x_i is one of the input signals, w_i is the weight for each connection, b is the bias, and $f(\bullet)$ represents an activation function such as binary step function,

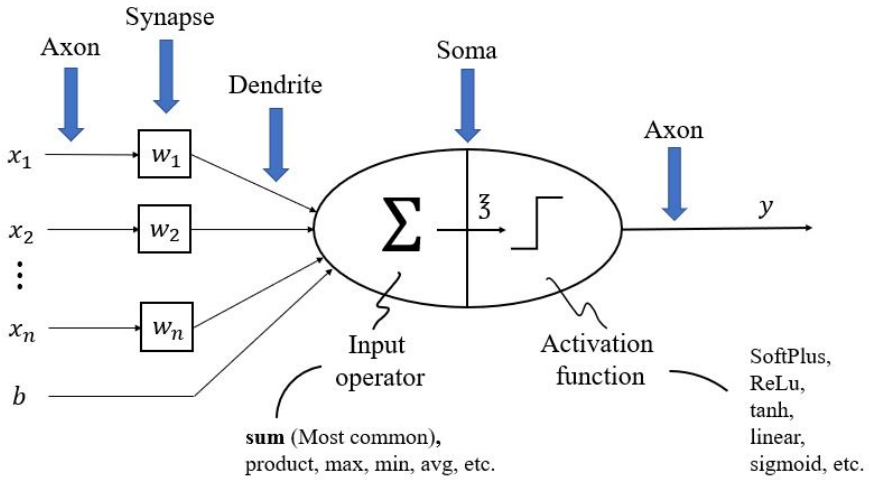


Figure 2.4: Neural network model for each neuron

sigmoid function, rectified linear unit (ReLU), softplus function, and etc.

2.2.2 Auto-associative neural networks (autoencoder)

The auto-associative neural network is a feedforward neural network with five layers: an input layer, an output layer, and three hidden layers [21]. Layer 1 is called the input layer, where the number of the nodes equals to the dimension of the input signals. The second layer is called the mapping layer, which generates the mapping from the original high-dimensional data to a higher dimensional space with the nonlinear mapping basis, then down to the third layer of low-dimensional principal components, which is named the bottleneck layer. After that, layer 4 produces a "demapping" from the low-dimensional principal components to the high-dimensional outputs. Thus, layer 4 is called the demapping layer. The last layer, layer 5, is the output layer which contains as many nodes as in layer 1. The topology for this special neural network is shown in Fig. 2.5.

It shows the example of auto-associative neural network with 3-6-2-6-3 layer. The

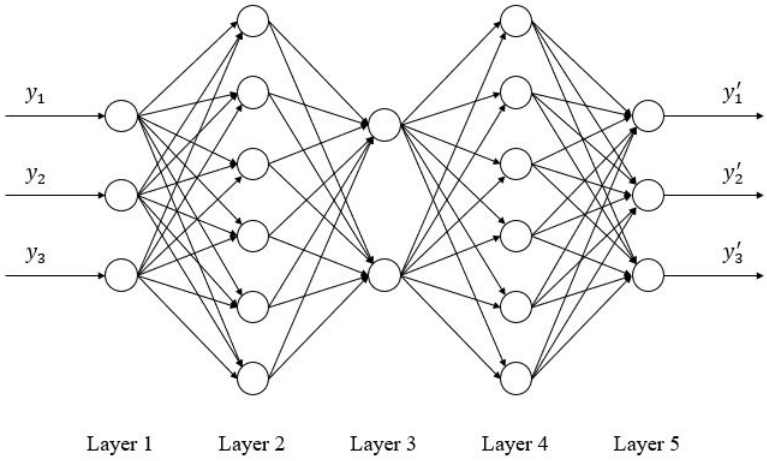
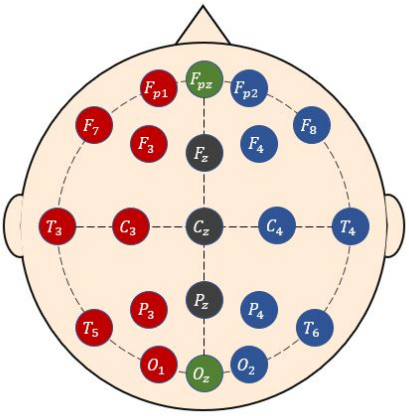


Figure 2.5: Schematic of autoencoder for a (3-6-2-6-3) encoding/decoding network. Modified from [21].

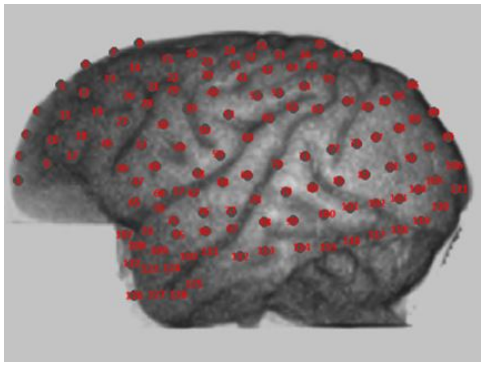
dimension of the input is three. From layer 1 to layer 3, two "principal components" is calculated via mapping layer with six hidden nodes. Such a process is called "encoding". From layer 3 to layer 5, the so-called "decoding" process estimates the outputs with the original dimension via demapping layer with six hidden nodes. This autoencoder is then trained by minimizing the errors between the inputs and the outputs.

2.2.3 Dimension matching for EEG and ECoG signals

From the experiments, we have 18 channels for EEG and 128 channels for ECoG. The placement of EEG electrodes was based on 10/20 system and the ECoG electrodes were distributed evenly over the left hemisphere of the brain. The location of the electrodes and those labels are shown in the Fig. 2.6. According to the geometry, EEG signals acquired from the left side of the brain are much more relevant to the ECoG signals. As a result, 10 EEG channels are chosen for further analysis: $F_{p1}, F_7, F_3, F_z, T_3, C_3, T_5, P_3, P_z$, and O_1.



(a) EEG 10/20 system. Modified from [24]



(b) Location of ECoG electrodes (from NeuroTycho).

Figure 2.6: Distribution and the labels of the recording electrodes

To match the dimension for EEG signals and ECoG signals, the dimension of ECoG must be reduced from 128 to 10. The stronger correlation between these two signals exists as the distance between two electrodes is nearer. Accordingly, n-nearest ECoG channels are assigned to each EEG channel. In order to cover all of the channels, n should be $\geq \lceil 128/10 \rceil = 13$. In the thesis, I choose $n = 16$ to avoid missing information. Based on the geometry, EEG channels and the corresponding ECoG channels are displayed in Table 2.1.

2.3 Results

The performance of the NLPCA can be analyzed by calculating the root mean square error between the output from the model and the actual signals.

$$R.M.S.E. = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{2.2}$$

Table 2.1: Labels of EEG channels and those corresponding 16 ECoG channels

EEG channel	Corresponding ECoG channel
F_{p1}	$1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 17, 18, 19, 26, 27, 36$
F_7	$36, 47, 48, 56, 57, 58, 65, 66, 67, 74, 75, 76, 96, 107, 108, 109$
F_3	$21, 27, 28, 29, 30, 37, 38, 39, 40, 48, 49, 50, 51, 58, 59, 60$
F_z	$6, 7, 8, 13, 14, 15, 16, 20, 21, 22, 23, 24, 28, 29, 30, 39$
T_3	$76, 77, 95, 96, 97, 109, 110, 111, 112, 122, 123, 124, 125, 126, 127, 128$
C_3	$24, 25, 31, 32, 33, 34, 35, 41, 42, 43, 44, 52, 53, 54, 61, 62$
T_5	$79, 80, 81, 82, 98, 99, 100, 101, 102, 103, 113, 114, 115, 116, 117, 118$
P_3	$60, 61, 62, 63, 69, 70, 71, 72, 73, 78, 79, 80, 81, 82, 99, 100$
P_z	$45, 46, 55, 63, 64, 72, 73, 83, 84, 85, 86, 87, 88, 89, 90, 92$
O_1	$88, 89, 90, 91, 92, 93, 94, 102, 103, 104, 105, 106, 118, 119, 120, 121$

where, N represents the total number of the samples, $\hat{y}$ is the reconstructed signals after the NLPCA model, and y represents the actual signals. Since each model for a EEG channel can reconstruct 16 channels of the ECoG signals, I will calculate the $R.M.S.E.$ for each channel, combine all 16 channels' outcomes together, and then visualize the results using a box plot.

From Fig. 2.7, it is obvious that the NLPCA model for EEG channel: $C3$ is the best one among all of the channels. Thus, I use the model obtained at $C3$ channel and apply it in all channels, in an attempt to determine the possibility of developing a generalized model by only one channel's model. The comparison with Fig. 2.7 is shown in Fig. 2.8.

When looking into both Fig. 2.7 and Fig. 2.8, we note that using C_3 only does not predict accurately. Hence, in order to have more accurate estimation, it is better to have a NLPCA model for each channel. In addition, I discovered that the error tends to be larger in some channels. I realized that root mean square of error does not take the magnitude of the signal into consideration. Hence, a new error

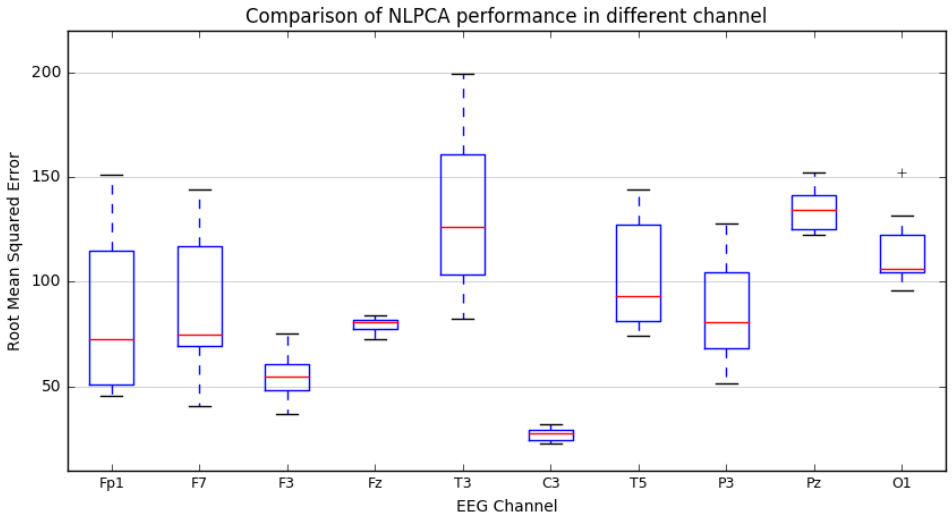


Figure 2.7: Box plot of the root mean square error for each channel. Red line represents Median (50th percentile), blue box represents 50% of values from 25th to 75th percentile, black line represents 5th percentile (bottom one) or 95th percentile (top one), and + represents the outliers.

function E is introduced:

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{(\hat{y}_i - y_i)^2}{1+\hat{y}_i{}^2} \tag{2.3}$$

when

$$\begin{cases} \hat{y} \gg 1 & \Longrightarrow \quad relative\ error \\ \hat{y} \approx 1 & \Longrightarrow \quad absolute\ error \end{cases} \tag{2.4}$$

It provides more suitable comparison when the signals cover large range of amplitude. The performance comparison is shown in Fig. 2.9.

All the root mean squared error, new error measure, and its standard deviation of each channel are calculated and shown in Table A.1 in Appendix. The root mean squared error ranges from 22.7299 to 199.4430; while the new error measure ranges from 0.0696 to 1008.2533.

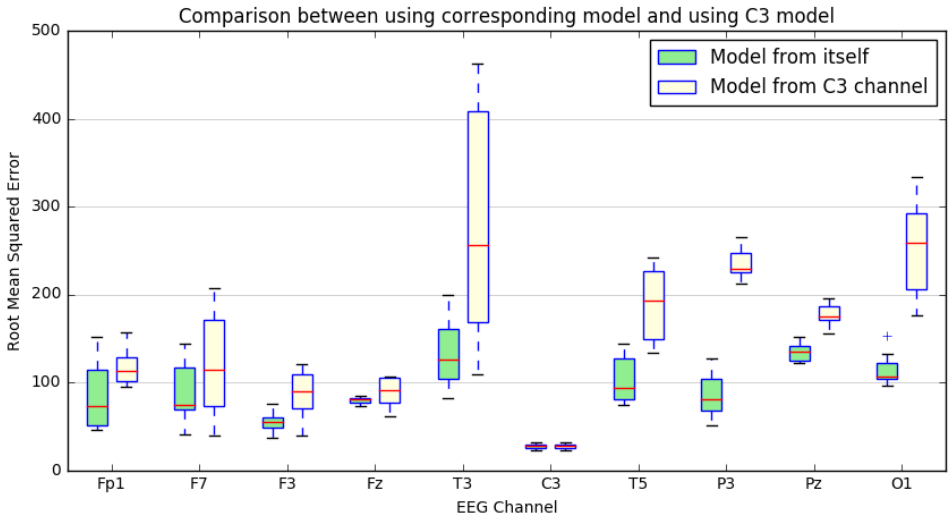


Figure 2.8: Box plot of the root mean square error using $C3$ NLPCA model for each channel. The error becomes larger when using the model from disparate channel, which means that the performance is getting worse if the model is not developed from itself. As expected, F_3, F_z, P_z which are close to C_3 show relative good results. What is surprising is that T_3, which is also close to C_3 has the largest error.

Finally, since the total number of ECoG channels is 128 and it is not very useful to have all the plots in the thesis, I would only display three of them (with the lowest E, the largest E, and an average E) to show how accurate the model can achieve. These are shown Fig. 2.10 through Fig. 2.12.

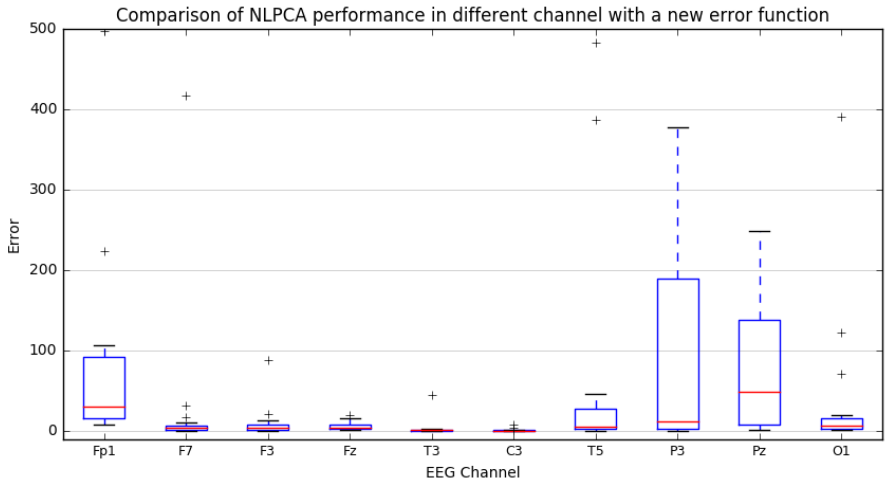


Figure 2.9: Box plot of the new error value for each channel.

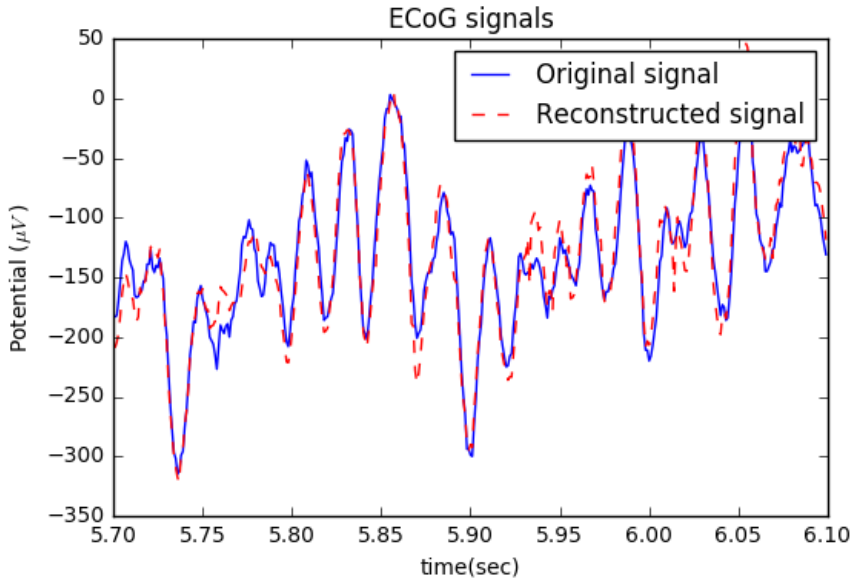


Figure 2.10: **Best Case:** Comparison between original signal and reconstructed signal from time 5.70 to 6.10 (sec) at ECoG channel: 53 using EEG channel: $C3$ model. This channel has the best NLPCA performance since the original signal and reconstructed signal match with each other nearly perfect, where the error and its standard deviation are 0.0696, and 0.1162 respectively from Table A.1.

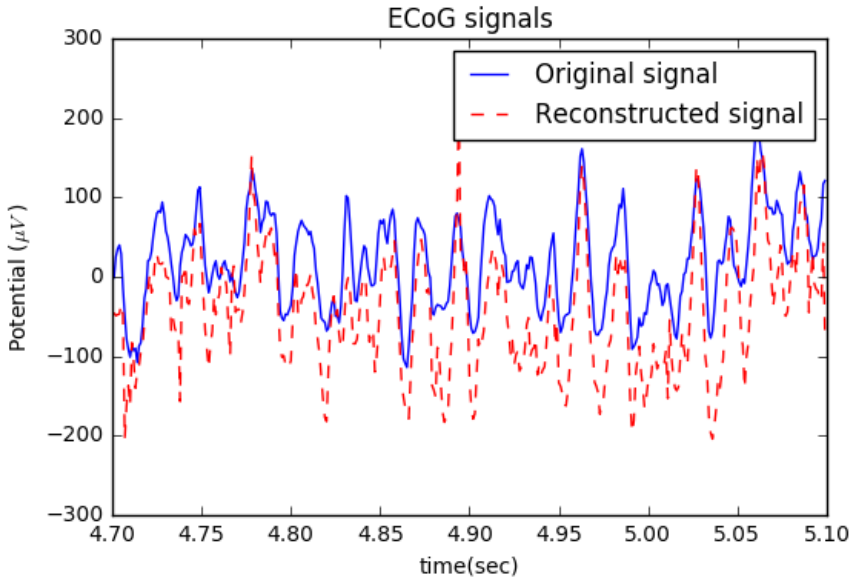


Figure 2.11: **Worst Case:** Comparison between original signal and reconstructed signal from time 4.70 to 5.10 (sec) at ECoG channel: 26 using EEG channel: F_{p1} model. This channel has the worst NLPCA performance since the the model has different estimation from the original one, where the error and its standard deviation are 1008.2533, and 3441.2676 respectively from Table A.1.

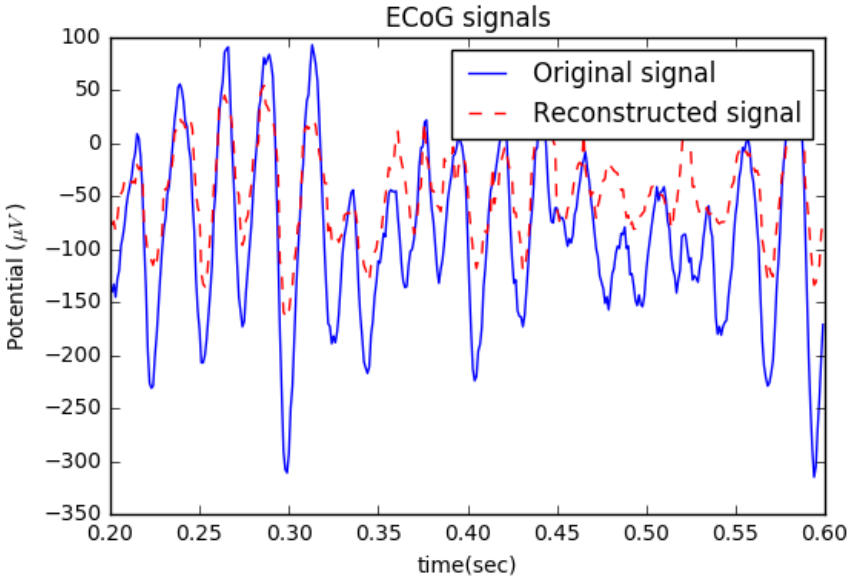


Figure 2.12: **Average Case:** Comparison between original signal and reconstructed signal from time 0.20 to 0.60 (sec) at ECoG channel: 70 using EEG channel: $P3$ model. This channel has the average NLPCA performance, where the error and its standard deviation are 65.4373, and 235.1448 respectively from Table A.1.

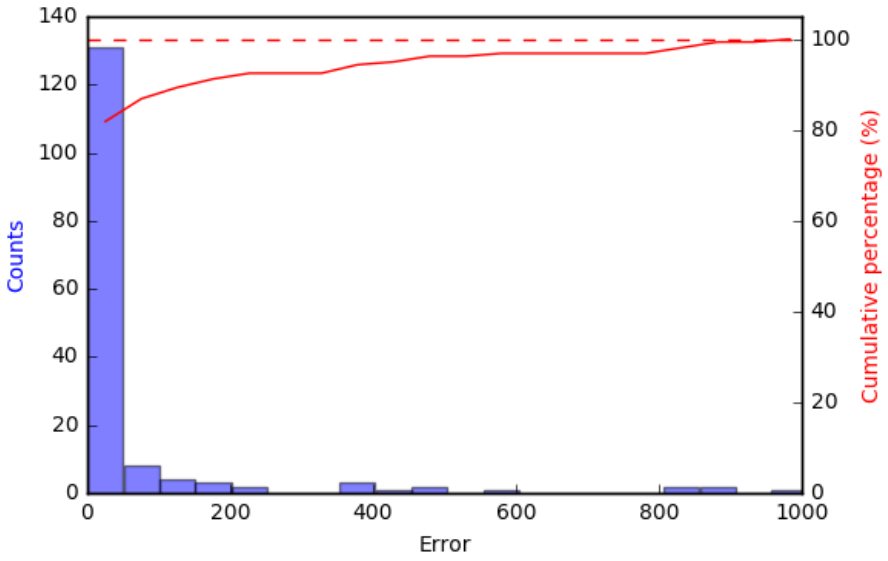


Figure 2.13: Distribution and cumulative percentage of the new error measure with bin width 50. Nearly 130 channels (nearly 80% of all the channels) have lower error than the average case.

As we can see from the figures, especially Fig. 2.13, we have a pretty accurate estimation for most of the channels. Even for the worst one in Fig. 2.11, some of the shape of pattern dynamics can be recovered well. And this information serves as a key part for further application such as neurological disorder diagnosis or mind controlling technique.

Chapter 3

Solution for Forward & Inverse Problem

3.1 Electrophysiology-based forward neural network model

3.1.1 Physiology of brain tissue and its effect on electrical signal flow

EEG signals are recorded on the scalp while ECoG signals are measured on the surface of the dura mater. Thus, the layers between the recording location of EEG and ECoG are skull and scalp. Assuming that the ECoG recordings are the electrical source, the EEG is the measuring of the voltage, and the effects of skull and scalp represent the effective impedance, we can model it as an electrical circuit.

The skull is the bony structure that protects the human brain from injury. It contains sinus cavities and numerous foramina to lessen the weight of the skull. From the anatomy, the skull can be seen as a structure consisting of three layers: a spongy bone layer in the midst of two compact bone layers. Thus, when modeling the skull effect, we will regard it as three layers. The middle layer has numerous cavities to

reduce the weight. The holes impedance can be modeled as capacitive that provide a potential field inside them. The impedance of the trabeculae is mainly resistive.

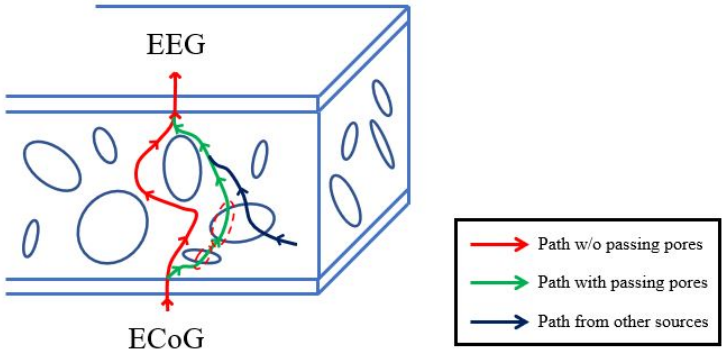


Figure 3.1: Schematic of electrical signal pathway through skull

There are various path for the electrical signals to travel. The first possible pathway is the path that does not pass any pore, where only resistance is included. The second path does pass through some pores, where resistance and capacitance are included. The last possible transmission is the path from other neighboring resources, where resistant and capacitance are included. This is also a network of elements laterally that also affect the signals propagation. Figure 3.2 shows a simplified electrical model suitable to study signal propagation.

When modeling the scalp effect, although it consists of several layers: skin, connective tissue, epicranial aponeurosis, loose areolar connective tissue, and pericranium, I model those as an effective resistance and a capacitance in parallel due to its relatively small effect on electrical signal transmission. Combining together, EEG-ECoG transmission electrical circuit is shown in Fig. 3.3.

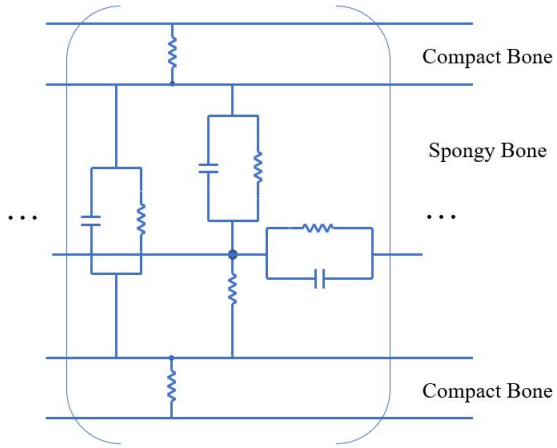


Figure 3.2: Electrical circuit for skull

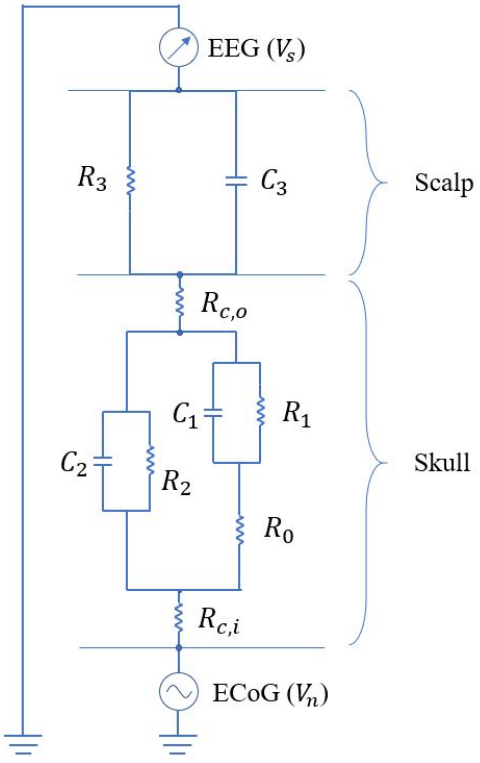


Figure 3.3: Signal transmission circuit for forward model. The bottom part represents the skull (see Fig. 3.2) and the top part is the model of the scalp.

where, ECoG signal and EEG signal are regarded as an electrical source and a voltmeter respectively.

3.1.2 Ordinary Differential Equations (ODEs) based on Neural-Bond-Graph (NBG) for forward dynamics

Bond graph (BG) [25] is a powerful tool for modeling physical systems. BG modeling is based on energy and energy exchange. Accordingly, it is widely employed in modeling engineering system with different physical domains such as electrical, mechanical, thermodynamic, hydraulic, and so on. Moreover, BG makes the complex system easier to be modeled and the ODEs for the system can be easily derived.

With the effective electrical circuit for electrical signal pathways shown in Fig. 3.3, its Bond Graph is generated as shown in Fig. 3.4:

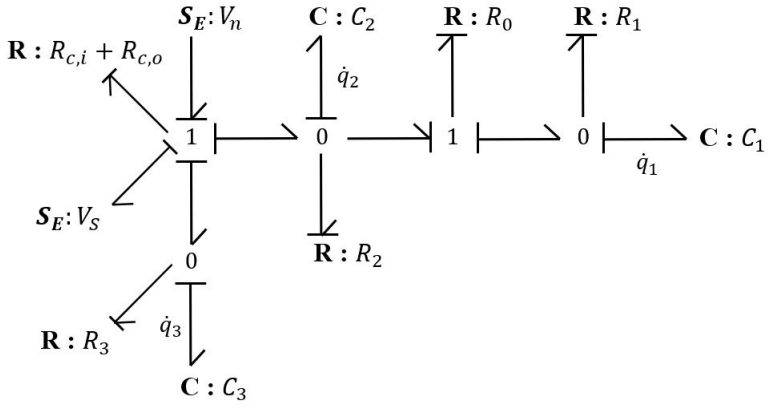


Figure 3.4: Bond Graph Model for forward problem. The two sources are the ECoG potential, V_n, and the EEG potential, V_s.

where, the charges q are represented the states when deriving the ODEs.

Given the causality assignments, the system is 3^{rd} order. Based on Fig. 3.4, the equations for the system is derived as follow:

$$\begin{cases} \dot{q}_1 = \frac{1}{R_0}(\frac{q_2}{C_2} - \frac{q_1}{C_1}) - \frac{1}{R_1}(\frac{q_1}{C_1}) \\ \dot{q}_2 = \frac{1}{R_{c,i}+R_{c,o}}(V_n - \frac{q_3}{C_3} - V_s - \frac{q_2}{C_2}) - \frac{1}{R_2}(\frac{q_2}{C_2}) - \frac{1}{R_0}(\frac{q_2}{C_2} - \frac{q_1}{C_1}) \\ \dot{q}_3 = \frac{1}{R_{c,i}+R_{c,o}}(V_n - \frac{q_3}{C_3} - V_s - \frac{q_2}{C_2}) - \frac{1}{R_3}(\frac{q_3}{C_3}) \end{cases} \tag{3.1}$$

While equation 3.1 represents the ideal condition where the resistance R and the capacitance C are linear. In case of uncertainty and continuous changing of human's brain tissue, nonlinearity is introduced in the equation:

$$\begin{cases} \dot{q}_1 = F_{R_0}^{-1}(F_{C_2}^{-1}(q_2) - F_{C_1}^{-1}(q_1)) - F_{R_1}^{-1}(F_{C_1}^{-1}(q_1)) \\ \dot{q}_2 = F_{R_{c,i}+R_{c,o}}^{-1}(V_n - F_{C_3}^{-1}(q_3) - V_s - F_{C_2}^{-1}(q_2)) - F_{R_2}^{-1}(F_{C_2}^{-1}(q_2)) \\ \qquad\qquad -F_{R_0}^{-1}(F_{C_2}^{-1}(q_2) - F_{C_1}^{-1}(q_1)) \\ \dot{q}_3 = F_{R_{c,i}+R_{c,o}}^{-1}(V_n - F_{C_3}^{-1}(q_3) - V_s - F_{C_2}^{-1}(q_2)) - F_{R_3}^{-1}(F_{C_3}^{-1}(q_3)) \end{cases} \tag{3.2}$$

$$V_s = V_n - F_{R_{c,i}+R_{c,o}}(\dot{q}_3 + F_{R_3}^{-1}(F_{C_3}^{-1}(q_3))) - F_{C_3}^{-1}(q_3) - F_{C_2}^{-1}(q_2) \tag{3.3}$$

where $F(\dots)$ is a nonlinear function to be determined.

To approximate the unknown nonlinear relationship, Artificial Neural Networks (ANNs) technology is introduced to incorporate the knowledge obtained from the pattern recognition capabilities of neural network and the physical information about the system through BG. The architecture of NBG is a sparse recurrent Multi-Layer Perception (MLP) network where it can identify the nonlinear dynamics recursively [26].

From the ODEs in equation 3.2, a Neuro-Bond-Graph (NBG) can be generated as follow:

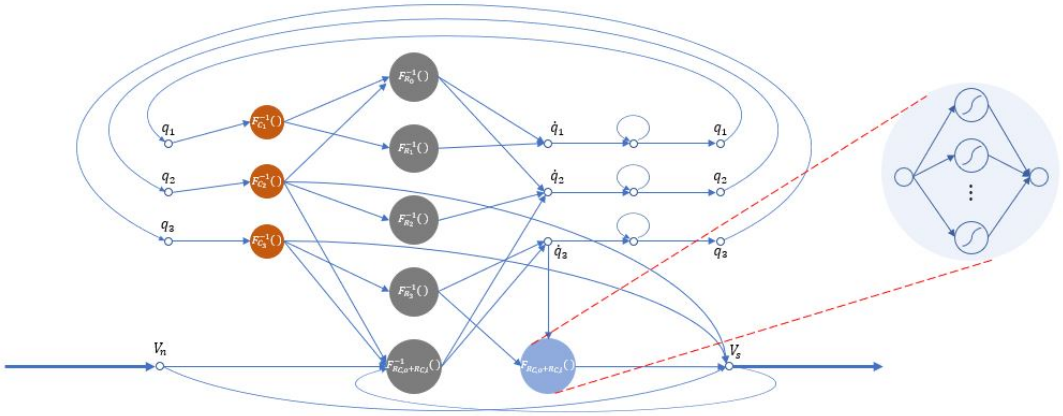


Figure 3.5: Neuro-Bond-Graph for forward problem. Note that the NBG is a sparse network that maps topology of the ODEs in (3.2) and (3.3).

3.1.3 Recurrent neural network (RNN) modeling & Long Short Term Memory Networks

From the equation 3.2, the model should include three characteristics: nonlinearity, recurrence (the connection from output to the hidden node) and hidden dynamic states (q, and $\dot{q}$). Nonlinear AutoRegressive network with eXogenous inputs (NARX) [27] is a good fit for forward model. The mathematical expression for NARX is given as follow:

$$y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_y), u(t), u(t-1), \ldots, u(t-n_u)) \tag{3.4}$$

where, y represents the output, u represents the input, n_y and n_u represent the number of the output and input delay respectively, and f is a nonlinear function of current input, delayed inputs and delayed outputs.

The feedforward neural network is employed to approximate the nonlinear function f. Then NARX neural network is displayed in Figure 3.6.

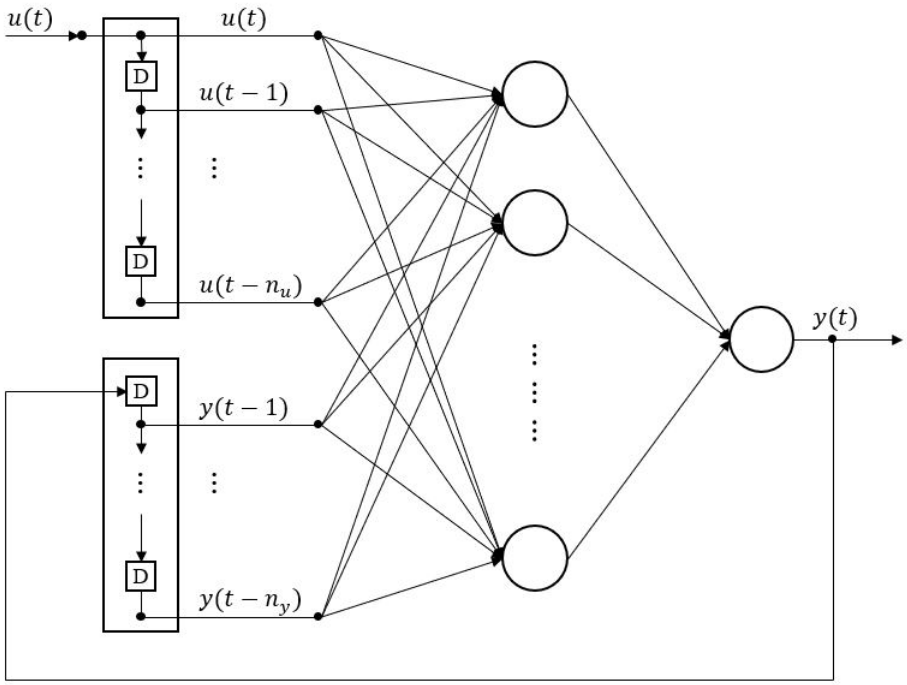


Figure 3.6: Nonlinear AutoRegressive neural network with eXogenous input (NARX) model. Modified from [27].

Based on the schematic of the NBGs, n_y and n_u equal to 1.

In addition, the model have loops. We can see the chain-like visualization when we unroll the loop in Fig. 3.7. Such a model performs well when the data only possess short-term dependencies. If the gap is too large, such a standard RNN becomes unable to learn to connect those pieces of information, resulting in poor model performance. However, in theory, RNN should be able to handle such "long-term dependencies". Thanks to Hochreiter & Schmidhuber's work [28], Long Short Term

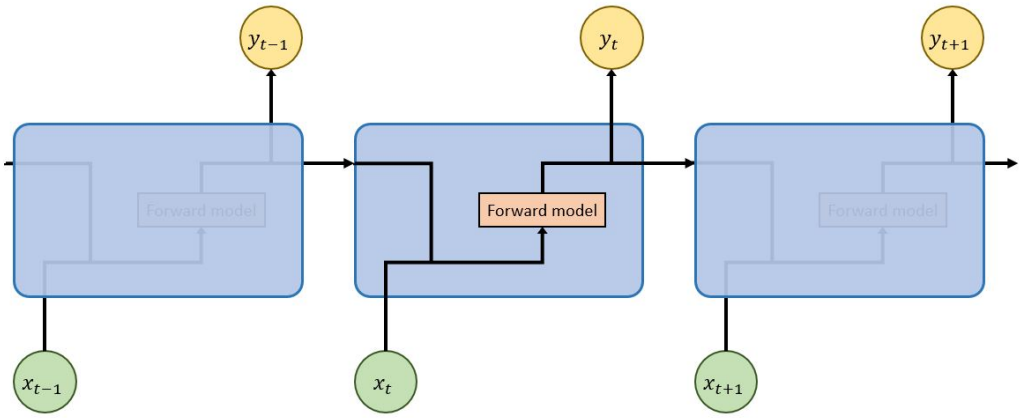


Figure 3.7: Unrolled RNN shows the chain-like nature. Modified from [29]

Memory networks (LSTMs) are introduced to deal with this issue.

LSTMs have special mechanisms, allowing it to remember the information for long periods of time as its natural behavior rather than something they struggle to learn. For the simple RNN in Fig. 3.7, the repeating module is just feeding the hidden state and the input to the forward model to obtain the current output. For LSTMs, the repeating module is something much more complicated (see Fig. 3.8).

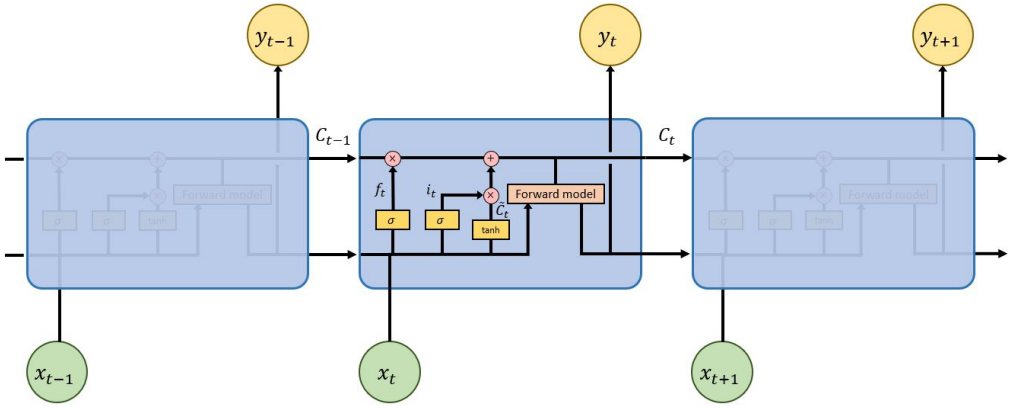


Figure 3.8: Long Short Term Memory Networks model. Modified from [29]

From the diagram in Fig. 3.8, there exists a cell state which is the horizontal line going through the top of the diagram. It conveys the useful information straight down the entire chain, with only some minor linear interactions. The next step is to determine what information we are going to throw away from the cell state by a sigmoid layer, forget gate layer.

$$f_t = \sigma(w_f \cdot [y_{t-1}, x_t] + b_f) \tag{3.5}$$

It takes the previous output, y_{t-1}, and the current input, x_t, into consideration to make the decision. The output value, f_t, is between 0 and 1.

Then, the next step is to decide what new information should be stored in the cell state. First, the decision of which values will be updated is made by another sigmoid layer, input gate layer.

$$i_t = \sigma(w_i \cdot [y_{t-1}, x_t] + b_i) \tag{3.6}$$

A vector of new candidate values, $\tilde{C}_t$, to be added to the cell state is then created by a tanh layer.

$$\tilde{C}_t = tanh(w_c \cdot [y_{t-1}, x_t] + b_c) \tag{3.7}$$

Based on the previous steps, we are able to update the old cell state, C_{t-1}, into a new one, C_t. Forget the things decided to be forgotten by multiplying the old state by f_t. Then store the new memory by adding the new candidate information scaled by how much decided to be updated, $i_t * \tilde{C}_t$.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.8}$$

Eventually, we take hidden state, C_t, previous output, y_{t-1}, and current input, x_t, as inputs to the forward model, we are able to generate current output, y_t, for each time step.

3.1.4 Results

Performance analysis of the forward model integrating with LSTMs is performed the same way as in Chapter 2: calculate the root mean squared of the error, another error function and its standard deviation, and then make a comparison. Here, we show the results of the forward model with the best, the worst, and the average estimation respectively. The distribution and the cumulative percentage of the new error are displayed as well in Fig. 3.12.

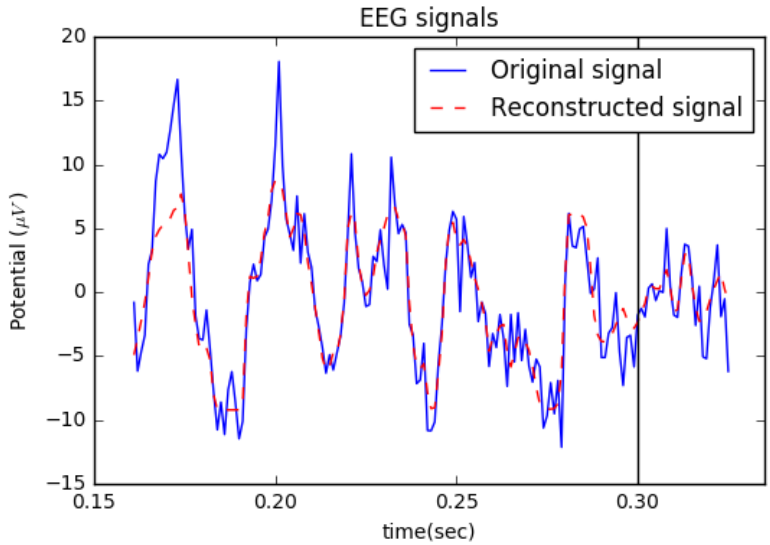


Figure 3.9: **Best Case:** Comparison between original signal and reconstructed signal from time 0.16 to 0.30 (sec) (with training data) and from time 0.30 to 0.33 (sec) (without training data) at EEG channel: $C3$. This channel has the best forward model performance. Although few peaks cannot be recovered, the reconstructed signal and further prediction still capture most of the dynamics of the signal, where the error and its standard deviation are 0.7981, and 2.4571 respectively from Table A.2.

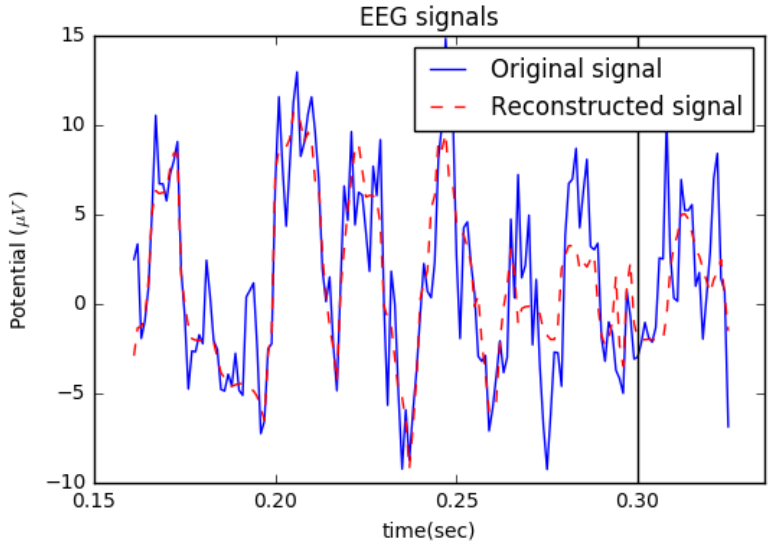


Figure 3.10: **Worst Case:** Comparison between original signal and reconstructed signal from time 0.16 to 0.30 (sec) (with training data) and from time 0.30 to 0.33 (sec) (without training data) at EEG channel: Fz. This channel has the worst forward model performance since the model is not able to capture most of the dynamics of the original signal, where the error and its standard deviation are 1.5306, and 6.0478 respectively from Table A.2.

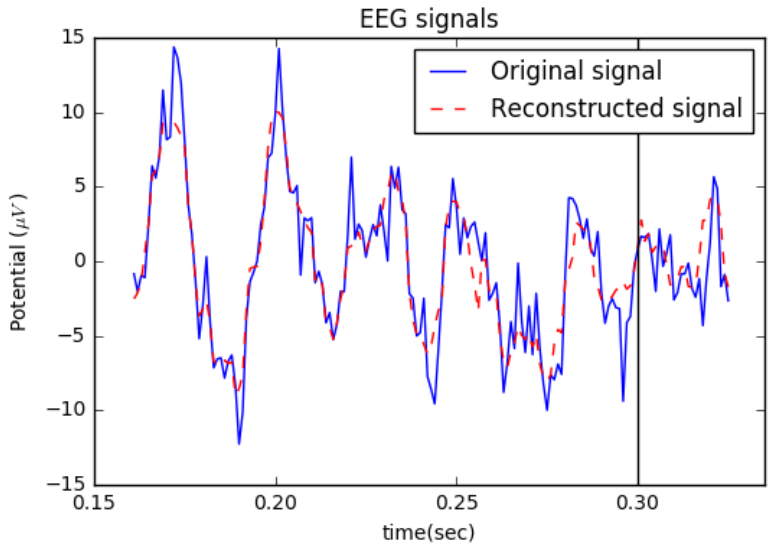


Figure 3.11: **Average Case:** Comparison between original signal and reconstructed signal from time 0.16 to 0.30 (sec) (with training data) and from time 0.30 to 0.33 (sec) (without training data) at EEG channel: $T5$. This channel has the average forward model performance. Further prediction can capture some of the dynamics of the signals, where the error and its standard deviation are 1.0541, and 3.1589 respectively from Table A.2.

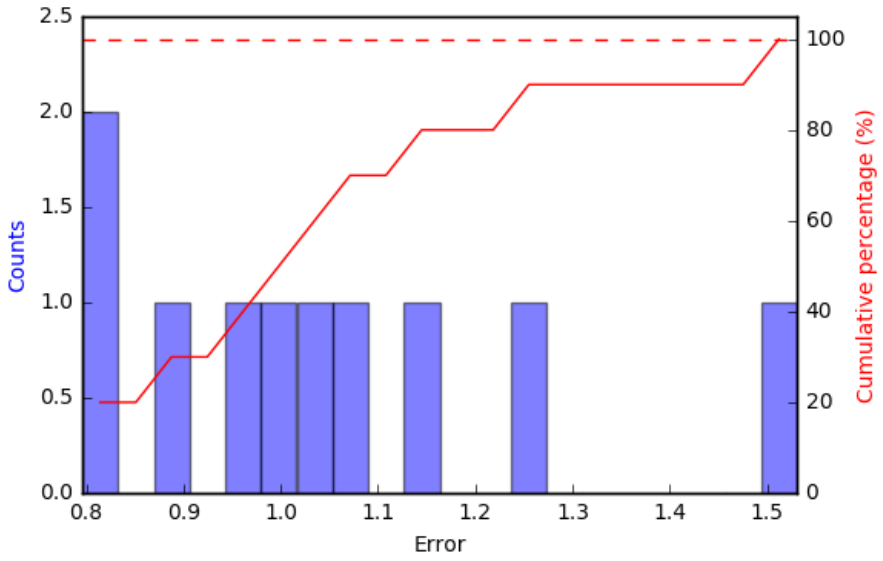


Figure 3.12: Distribution and cumulative percentage of the new error measure with bin width 0.037. Nearly 5 channels (nearly 50% of all the channels) have lower error than the average case.

From Fig. 3.9 though Fig. 3.11, we can see that the higher order of the dynamic is hard to be reconstructed. It may result from the noisy measurement during the experiments, that I didn't take into account. However, for the other parts of the data, it has been shown that those features of the signals can be estimated well by the forward model developed in this chapter.

3.2 Inverse model

3.2.1 LSTM, Multi-Layer Perceptron (MLP) & Inverse NLPCA

For the inverse solution, I employed a multi-layer perceptron to develop a deep neural network (DNN) model, which is used to capture the dynamics of the signals. On top of that, a LSTM is included for recurrent connection. After obtaining the estimated principal components, there exists the de-mapping part of the Auto-Associative

Neural Network, from Layer 3 to Layer 5 in Figure 2.5. The ECoG signals can then be estimated. Thus, the inverse model, which estimates ECoG signals from EEG signals, is considered as the combination of MLP and de-mapping network from NLPCA.

3.2.2 Multi-Model determination

When selecting 16 ECoG channels for each EEG location, there exists overlapping areas as shown in Table 3.1, where multiple models can estimate the same channels for ECoG. Given different estimations from different models, we select the model with better performance to determine the ultimate estimation for inverse solution.

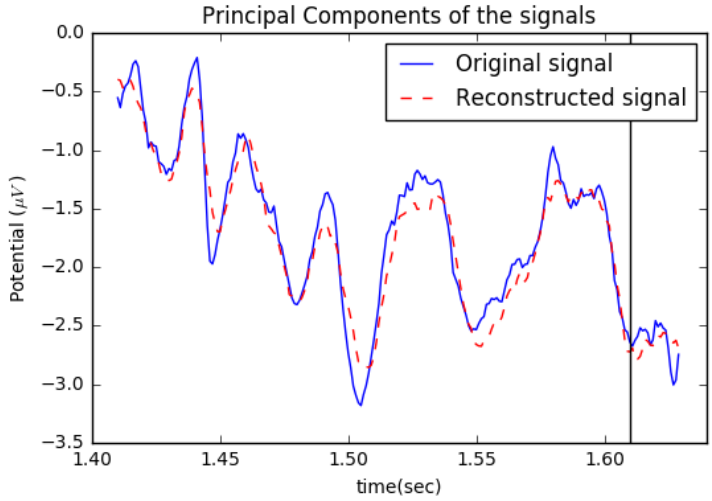
Table 3.1: Overlapping EEG channels corresponding to ECoG channels. Bold text represents the EEG channel model with better performance.

ECoG	Overlapping EEG channel	ECoG	Overlapping EEG channel
6	$\mathbf{F_{p1}}$ & F_z	76	$\mathbf{F_7}$ & T_3
21	F_3 & $\mathbf{F_z}$	79	$\mathbf{P_3}$ & T_5
24	$\mathbf{F_z}$ & C_3	80	$\mathbf{P_3}$ & T_5
27	F_{p1} & $\mathbf{F_3}$	81	P_3 & $\mathbf{T_5}$
28	F_3 & $\mathbf{F_z}$	82	P_3 & $\mathbf{T_5}$
29	F_3 & $\mathbf{F_z}$	88	$\mathbf{P_z}$ & O_1
30	F_3 & $\mathbf{F_z}$	89	$\mathbf{P_z}$ & O_1
36	F_{p1} & $\mathbf{F_7}$	90	$\mathbf{P_z}$ & O_1
39	$\mathbf{F_3}$ & F_z	92	P_z & $\mathbf{O_1}$
48	F_7 & $\mathbf{F_3}$	96	F_7 & $\mathbf{T_3}$
58	F_7 & $\mathbf{F_3}$	99	$\mathbf{T_5}$ & P_3
60	F_3 & $\mathbf{P_3}$	100	T_5 & $\mathbf{P_3}$
61	$\mathbf{C_3}$ & P_3	102	T_5 & $\mathbf{O_1}$
62	$\mathbf{C_3}$ & P_3	103	T_5 & $\mathbf{O_1}$
63	$\mathbf{P_3}$ & P_z	109	F_7 & $\mathbf{T_3}$
72	P_3 & $\mathbf{P_z}$	118	$\mathbf{T_5}$ & O_1
73	P_3 & $\mathbf{P_z}$		

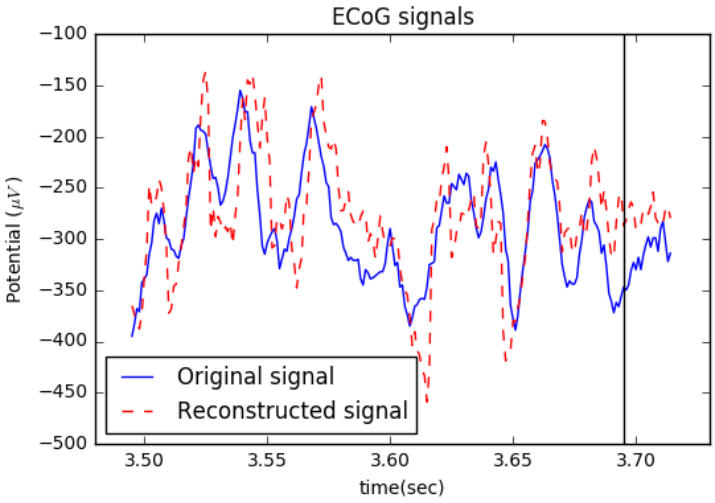
3.2.3 Results

First, I compared the estimated principal components with the actual principal components at each EEG channel. Then the performance evaluation of the inverse solution was performed by the comparison between the estimated ECoG signals and the true ECoG signals. Figure 3.13 through 3.15 show again a sample (Best, Average, Worst) of the results. The distribution and the cumulative percentage of the new error are also plotted in Fig. 3.16.

As we can see from the Fig. 3.13a, 3.14a, 3.15a, and 3.16a, the principal components can be recovered well, even in the worst case. There are only few misinterpretations. For the ECoG estimation from inverse model (Figure 3.13b, 3.14b, 3.15b, and 3.16b), the overall performance is worse than that of the principal components. Although the best case provides an accurate estimation, for the cases below the average, the results are not reliable. The main reason is the signal itself. The best correlation was seen at the Pz site which may reflect the most robust primary motor signal, even in a restrained animal. During the experiment, the monkey was under full general anesthesia where the subject had its eyes closed, both arms restrained. The effects of a general anesthetic is to dampen the amplitude of the signal. Furthermore, the EEG pattern under deep sleep or general anesthetic is slow Delta rhythm. This may have contributed to the difficulty in correlation of signals. Another reason is the de-mapping part of NLPCA. It has been shown in Chapter 2 that NLPCA cannot reconstruct the actual signals perfectly. The ultimate estimation would become even worse if the input feed to the system is not accurate. Also, I observed that the inverse solution seems to be either overfitting or underfitting. It might be due to either the connection between hidden nodes in the model is redundant or they compensate each other. Therefore, making the model partially connected by introducing electro-physiological properties is the way to deal with this issue.

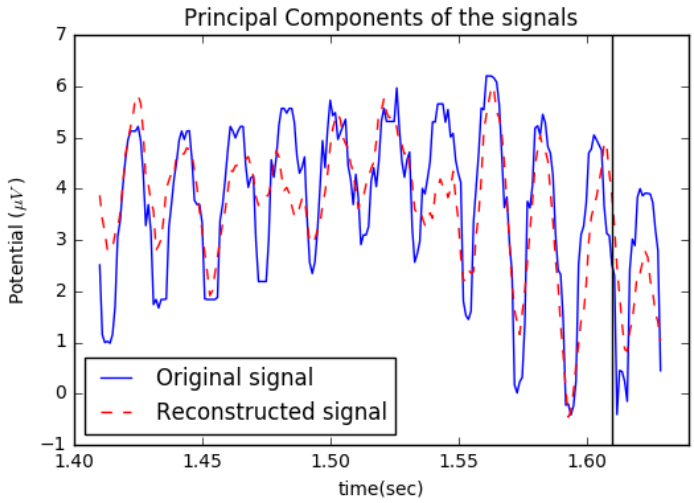


(a)

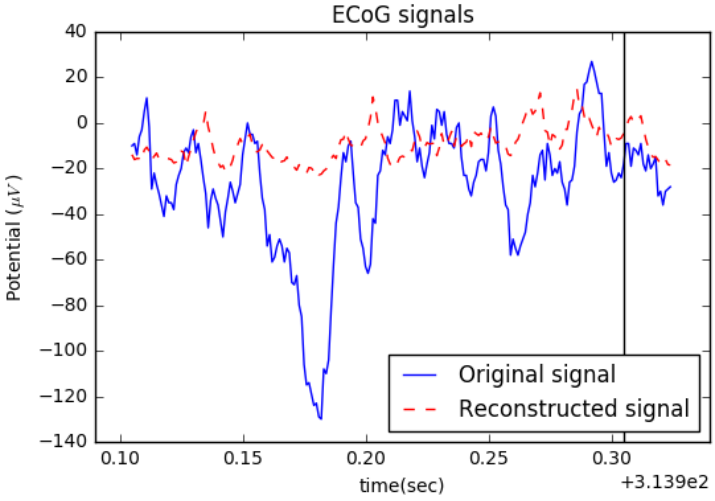


(b)

Figure 3.13: **Best Case:** Comparison between actual principal components (PCs) of the ECoG signals and the actual ECoG signals. (a) is the comparison from time 1.41 to 1.61 (sec) (with training data) and from time 1.61 to 1.63 (sec) (without training data) at the 5^{th} PC of the EEG channel: $P3$. This one has the best PC estimation performance where the error and the standard deviation are 0.0388 and 0.0883 respectively. (b) is the comparison from time 3.49 to 3.69 (sec) (with training data) and from time 3.69 to 3.71 (sec) (without training data) at ECoG channel: 84 using EEG channel: Pz. This one has the best inverse estimation performance where the error and the standard deviation are 18.1474 and 387.9468 respectively.

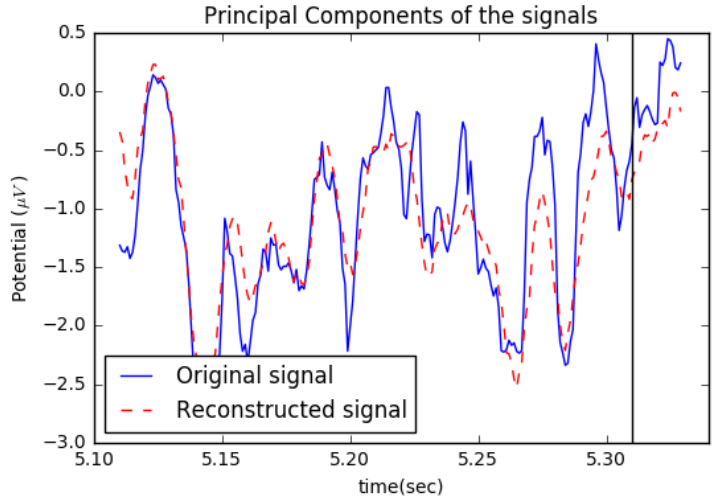


(a)

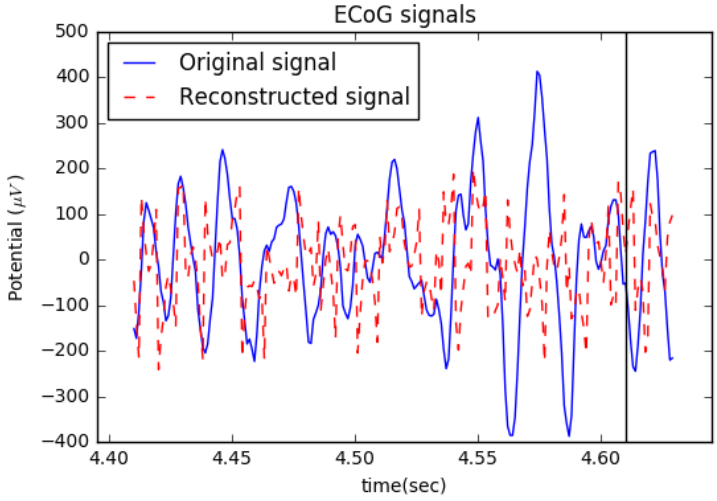


(b)

Figure 3.14: **Worst Case:** Comparison between actual principal components (PCs) of the ECoG signals and the actual ECoG signals. (a) is the comparison from time 1.41 to 1.61 (sec) (with training data) and from time 1.61 to 1.63 (sec) (without training data) at the 1^{st} PC of the EEG channel: $O1$. This one has the worst PC estimation performance where the error and the standard deviation are 1.1516 and 2.8391 respectively. (b) is the comparison from time 314.00 to 314.20 (sec) (with training data) and from time 314.20 to 314.22 (sec) (without training data) at ECoG channel: 71 using EEG channel: $P3$. This one has the worst inverse estimation performance where the error and the standard deviation are 1735.6951 and 9390.8913 respectively.

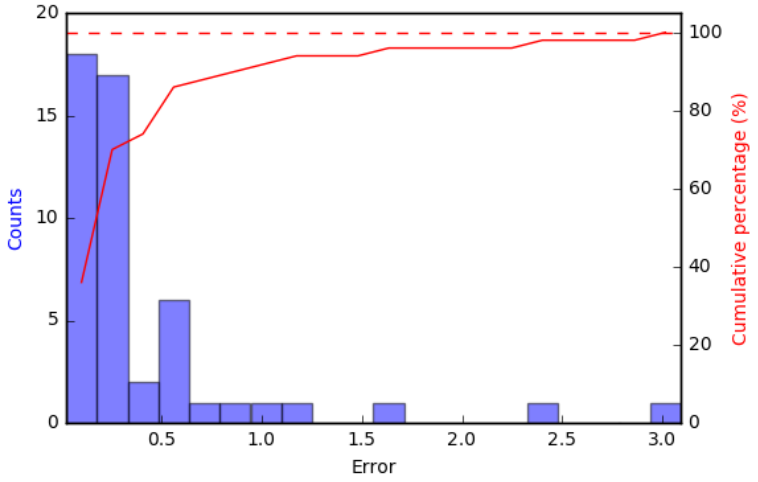


(a)

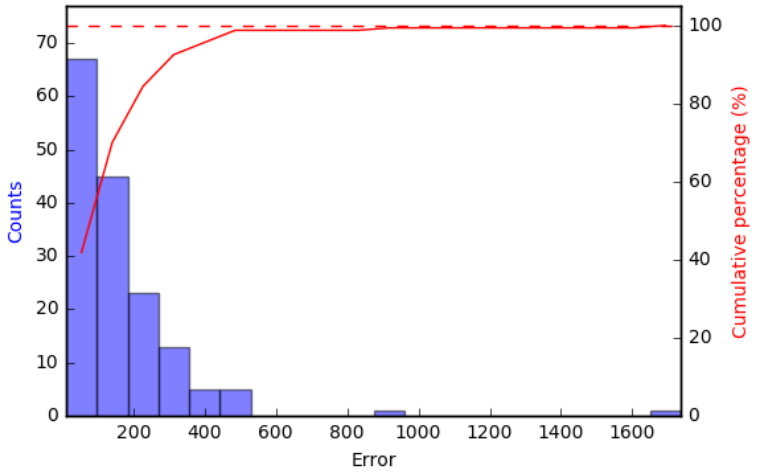


(b)

Figure 3.15: **Average Case:** Comparison between actual principal components (PCs) of the ECoG signals and the actual ECoG signals. (a) is the comparison from time 5.11 to 5.31 (sec) (with training data) and from time 5.31 to 5.33 (sec) (without training data) at the 3^{rd} PC of the EEG channel: $F7$. This one has the average PC estimation performance where the error and the standard deviation are 0.2861 and 0.4656 respectively. (b) is the comparison from time 4.41 to 4.61 (sec) (with training data) and from time 4.61 to 4.63 (sec) (without training data) at ECoG channel: 6 using EEG channel: $Fp1$. This one has the average inverse estimation performance where the error and the standard deviation are 161.0342 and 1907.9436 respectively.



(a)



(b)

Figure 3.16: Distribution and cumulative percentage of the new error measure for inverse model evaluation. (a) is the distribution and cumulative percentage of the new error measure between actual principal components (PCs) of the ECoG signals with bin width 0.163. Nearly 34 channels (nearly 68% of all the channels) have lower error than the average case. (b) is the distribution and cumulative percentage of the new error measure between actual ECoG signals with bin width 86.209. Nearly 112 channels (nearly 70% of all the channels) have lower error than the average case.

Chapter 4

Future Work

4.1 Dynamic Inverse Model

Given the Multi-variable time varying Bond Graph model shown in Fig. 3.4, the output source $\mathbf{S_E} : V_s$ can be approximated as $\mathbf{R} : R_s$. Then the equations for the system is re-written as follow:

$$\begin{cases} \dot{q}_1 = \frac{1}{R_0}(\frac{q_2}{C_2} - \frac{q_1}{C_1}) - \frac{1}{R_1}(\frac{q_1}{C_1}) \\ \dot{q}_2 = \frac{1}{R_{c,i}+R_{c,o}+R_s}(V_n - \frac{q_3}{C_3} - \frac{q_2}{C_2}) - \frac{1}{R_2}(\frac{q_2}{C_2}) - \frac{1}{R_0}(\frac{q_2}{C_2} - \frac{q_1}{C_1}) \\ \dot{q}_3 = \frac{1}{R_{c,i}+R_{c,o}+R_s}(V_n - \frac{q_3}{C_3} - \frac{q_2}{C_2}) - \frac{1}{R_3}(\frac{q_3}{C_3}) \end{cases} \tag{4.1}$$

$$V_s = \frac{V_n - \frac{q_3}{C_3} - \frac{q_2}{C_2}}{R_{c,i} + R_{c,o} + R_s} R_s \tag{4.2}$$

Thus, the system can be then expressed as follow:

$$\dot{x} = Ax + Bu \qquad\qquad y = Cx + Du \tag{4.3}$$

where,

$$x = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T, \qquad u = V_n \quad , \qquad y = V_s$$

$$A = \begin{bmatrix} \frac{-1}{C_1}(\frac{1}{R_0} + \frac{1}{R_1}) & \frac{1}{R_0 C_2} & 0 \\ \frac{1}{R_0 C_1} & \frac{-1}{C_2}(\frac{1}{R_0} + \frac{1}{R_2} + \frac{1}{R_{c,i}+R_{c,o}+R_s}) & \frac{-1}{C_3}\frac{1}{R_{c,i}+R_{c,o}+R_s} \\ 0 & \frac{-1}{C_2}\frac{1}{R_{c,i}+R_{c,o}+R_s} & \frac{-1}{C_3}(\frac{1}{R_3}\frac{1}{R_{c,i}+R_{c,o}+R_s}) \end{bmatrix} \tag{4.4}$$

$$B = \begin{bmatrix} 0 & \frac{1}{R_{c,i}+R_{c,o}+R_s} & \frac{1}{R_{c,i}+R_{c,o}+R_s} \end{bmatrix}^T$$

$$C = \begin{bmatrix} 0 & \frac{-1}{C_2}\frac{R_s}{R_{c,i}+R_{c,o}+R_s} & \frac{-1}{C_3}\frac{R_s}{R_{c,i}+R_{c,o}+R_s} \end{bmatrix}$$

$$D = \frac{R_s}{R_{c,i}+R_{c,o}+R_s}$$

The inversion algorithm for multi-variable system were obtained by Sain and Massey [30].

$$\begin{aligned} \dot{y} &= C\dot{x} + D\dot{u} \\ &= CAx + CBu + D\dot{u} \end{aligned} \tag{4.5}$$

$$\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} C \\ CA \end{bmatrix} x + \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} \begin{bmatrix} u \\ \dot{u} \end{bmatrix} = Ox + M \begin{bmatrix} u \\ \dot{u} \end{bmatrix} \tag{4.6}$$

There exists a matrix $K = [\kappa_1 \quad \kappa_2]$ such that

$$KM = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.7}$$

Then,

$$K \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = KOx + KM \begin{bmatrix} u \\ \dot{u} \end{bmatrix} = KOx + u \tag{4.8}$$

Thus, the inversion of the system is derived as follow:

$$\begin{aligned} u &= -KOx + \kappa_1 y + \kappa_2 \dot{y} \\ \dot{x} &= Ax + Bu \\ &= Ax - BKOx + B\kappa_1 y + B\kappa_2 \dot{y} \\ &= (A - BKO)x + B\kappa_1 y + B\kappa_2 \dot{y} \end{aligned} \tag{4.9}$$

where, $\dot{y}$ can either be approximated by $(y_k - y_{k-1})/\Delta t$ or be modeled by unknown input observer (UIO) [31] to obtain a robust state estimation even without *a priori* knowledge about unknown input. After introducing the nonlinearity to the system, an inverse version of NueroBondGraph is established.

4.2 Mixture of Experts

For the inverse solution, the estimations for some channels are determined by more than one neural network model. Mixture of Experts (MoE) [32] is the most commonly used combining method that has been shown to improve performance in machine learning potentially.

Given there exists n "experts" (Expert Network in the figure) generating each estimation y_i respectively. Then the "manager" (Gating Network in the figure) would determine the probability g_i of picking expert i for final output, where

$$\sum_i g_i = 1 \tag{4.10}$$

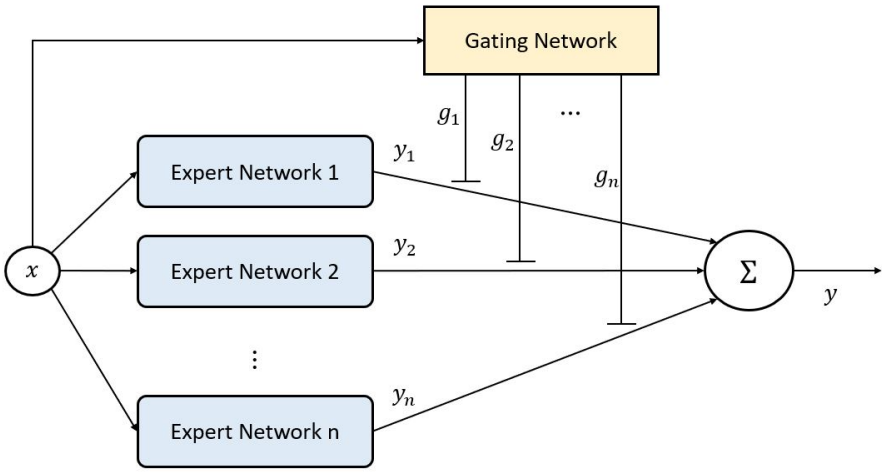


Figure 4.1: Schematic structure of Mixture of Experts (MoE)

Then the final output is given by

$$y = \sum_{i}^{n} g_i y_i \tag{4.11}$$

And the error function [33] for training the MoE is represented as:

$$E = -log \sum_{i} g_i e^{-\frac{1}{2}(y-y_i)^T(y-y_i)} \tag{4.12}$$

where it has been shown with better efficiency than just using squared of errors or squared of absolute errors.

4.3 Deep neural network pruning techniques

Although the model developed in the thesis is a sparse neural network, it is still very complex, especially when the number of hidden nodes exceeds 500. It is imaginable that there should exist over-fitting in the model. To improve the speed of learning and obtain better generalization, a neural network pruning technique such

as optimal brain damage (OBD) [34] and optimal brain surgeon (OBS) [35] can be employed to achieve this goal.

OBD can reduce the size of the neural network without a large decrease in the performance of the model by deleting the weights with small "saliency" [34], which means those whose deletion will have the least effect on the training error. First, we approximate the error function E by Taylor series with respect to a perturbation δU of the parameter:

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta {u_i}^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(\|\delta U\|^3) \tag{4.13}$$

where, h_{ij} is the element from the Hessian matrix H:

$$g_i = \frac{\partial E}{\partial u_i} \qquad\qquad h_{ij} = \frac{\partial^2 E}{\partial u_i \partial u_j} \tag{4.14}$$

To simplify the equation, two assumptions are introduced: (1) diagonal approximations assuming that δE caused by deleting several parameters equals to the sum of δE caused by each parameter so the cross term is ignored, and (2) extremal approximations assuming that the deletion is performed after training has converged so the term g_i equals to 0. Then the equation reduces to

$$\delta E = \frac{1}{2} \sum_i h_{ii} \delta {u_i}^2 \tag{4.15}$$

Then, calculate the saliency for each parameter: $s_k = h_{kk} {u_k}^2/2$, and eventually delete the parameters with low saliency.

An alternative approach that can be implemented to prune the neural network is OBS [35]. OBS and OBD are similar methods that delete the weights of the

network by the value of saliency calculated based on the Hessian matrix. However, there exists some differences. First, OBS does not introduce the diagonal approximation made in OBD. Second, derivation of saliency is based on the inverse Hessian matrix: $s_k = {u_k}^2/(2[H^{-1}]_{kk})$.

4.4 Implementation & demonstration on human data

The data analyzed in this thesis is from the open source: NeuroTycho [36]. EEG and ECoG signals were simultaneously recorded from a monkey with anesthetic drug injected sitting on a primate chair with its eyes closed and both arms restrained. We intend to apply this paradigm using human subjects at Dell Medical School at The University of Texas at Austin Hospital. Experiments will be designed to take advantage of EEG/ECoG simultaneous recording on epilepsy patients, who have undergone brain surgery.

During experiments, specific tasks will be designed to allow us to measure the event-related potentials (ERP) resulted from a sensory, cognitive, or motor activity [37]. For example, P300 (P3) wave is the ERP component that is related to the decision making. Sensory evoked potential such as visual, auditory, and somatosensory induced by environment stimulus is one of the most commonly measured ERP components as well. This event-locked signals would be essential to study the relationship between EEG and ECoG, and thus improve the inverse model solution. The ultimate goal is to demonstrate the improvement of BCI system performance due to the implementation of the inverse model developed in this thesis.

4.5 Feature prediction model

In Brain-Computer Interface systems, we will not just use raw signals from the electrodes recording to interpret the user's intent. Features extracted from both the time domain and the frequency domain play significant role on decoding the brain activities. For time series features, we can calculate a number of "features", such as the root mean square, mean absolute value, waveform length, an so on. Frequency analysis can be performed by applying Fast Fourier Transform (FFT). Mean frequency and power spectral density are the examples that can be extracted from frequency domain. Besides, signal analysis in both time and frequency domains simultaneously allows to explore more hidden information from the data. Such a time-frequency analysis is achieved by continuous wavelet transform (CWT) [38].

If those features extracted from ECoG signals f_{ECoG} can be estimated from the EEG signals, it is not necessary to employ the inverse model first to estimate ECoG and then transform the estimation into feature domain. Given the promising results of the inverse solution in the thesis, I will work on developing a feature prediction model, which input is EEG signals and the output is f_{ECoG}.

Appendix A

A.1 Analysis results of NLPCA

All the root mean squared error, new error measure, and its standard deviation of each channel between the true ECoG signals and the reconstructed ECoG signals from NLPCA are calculated and shown in Table A.1. The root mean squared error ranges from 22.7299 to 199.4430; while the new error measure ranges from 0.0696 to 1008.2533. Note that there is one channel: 68 with the value N/A since we discard this channel due to its location which is not in the effective cover range of EEG channels.

Table A.1: Results of NLPCA performance evaluation in three different parameters: root mean square of error (RMS), customize error (Error), and the standard deviation of Error (std.)

ECoG	RMS	Error	std.	ECoG	RMS	Error	std.
1	95.5227	29.1072	91.1017	65	137.8320	2.5320	6.5966
2	58.3043	20.3562	71.0600	66	144.2374	31.9541	120.0498
3	48.8462	9.1312	18.3242	67	140.4416	5.9893	15.8186
4	53.7451	86.5642	299.8310	68	N/A	N/A	N/A
5	48.4861	8.1418	19.0003	69	102.3846	10.3873	26.7394
6	81.7321	9.8082	29.2029	70	110.3444	65.4373	235.1448

Table A.1 Continued

7	77.0639	1.3121	1.1802	71	95.0714	32.7843	114.4496
8	78.4514	4.6378	10.9927	72	72.6717	0.9858	2.9017
9	45.2562	31.8049	60.0426	73	59.4873	0.5725	1.3204
10	51.5509	15.0742	40.4144	74	115.5470	5.9902	16.1810
11	63.8960	15.4535	39.7337	75	89.9539	3.3842	10.7966
12	81.6460	83.9387	273.5250	76	70.4769	0.5555	1.0817
13	81.0091	13.0545	23.8728	77	172.9431	1.5824	2.7351
14	81.6117	6.9419	14.9225	78	51.5555	194.1411	678.2894
15	82.4823	15.6926	55.5128	79	58.4848	0.2053	0.2769
16	80.5068	7.3140	21.4195	80	141.3165	13.4083	33.7453
17	110.1281	23.4010	40.8251	81	78.7179	1.5822	3.9779
18	128.2816	106.3411	248.8058	82	143.0945	12.1669	37.6524
19	142.7421	223.6099	689.8285	83	138.9358	198.9461	730.0501
20	75.5345	1.9737	4.1521	84	143.6693	63.2542	214.7185
21	72.8138	1.1881	2.0299	85	150.7082	34.2401	108.6031
22	74.4033	1.7245	3.3354	86	151.9958	7.5298	24.3813
23	77.4667	19.0332	42.7537	87	149.0155	4.9112	15.3987
24	22.7299	1.1176	2.2617	88	96.0065	2.3612	6.6547
25	24.8231	8.4112	31.4719	89	104.4723	4.7586	15.9006
26	151.2977	1008.2533	3441.2676	90	106.2260	5.6182	18.8307
27	56.8228	21.5151	47.2240	91	109.7958	6.6966	22.0549
28	80.8088	2.1313	2.8032	92	133.0163	1.4885	2.7705
29	80.8016	1.7757	2.6031	93	121.2846	19.7085	73.8247
30	83.9199	2.1146	3.0659	94	122.3034	391.0476	1511.9036
31	24.4960	1.5194	4.7585	95	199.4430	0.5266	0.8390

Table A.1 Continued

32	23.7812	3.3784	9.7367	96	195.6664	0.3816	0.8536
33	24.2877	0.3704	0.5487	97	176.0263	0.7608	2.5244
34	24.6488	0.2486	0.4288	98	125.0828	46.2557	167.3746
35	26.9440	0.1598	0.2592	99	111.2615	3.0281	7.7696
36	40.7614	2.3765	5.2636	100	95.7101	2.8538	9.2343
37	75.4289	9.2311	29.9298	101	81.2402	20.7032	79.8694
38	58.3741	1.5075	1.9567	102	122.4319	2.4384	6.7275
39	55.3589	0.7426	1.1288	103	76.0485	1.0230	3.3840
40	52.2632	0.4502	0.6818	104	126.6508	1.9921	6.1734
41	28.7489	0.1187	0.1841	105	152.4983	71.0843	252.4755
42	28.5805	0.0831	0.1221	106	104.2380	13.9211	51.6980
43	31.5625	0.0978	0.1542	107	69.8671	0.4702	0.9816
44	31.3804	0.0786	0.1164	108	73.6006	0.4540	0.8217
45	122.3759	8.5731	23.3579	109	71.4085	0.5520	1.2730
46	123.2699	84.3321	301.0731	110	137.4681	0.4244	1.3914
47	42.4503	1.0558	1.5458	111	127.0666	0.0839	0.1005
48	54.2262	0.4638	0.7797	112	108.4809	0.0946	0.1268
49	53.5484	0.4560	0.7621	113	80.2778	0.3322	0.7315
50	48.7027	0.3574	0.5859	114	85.6397	0.5329	1.8124
51	47.7265	0.6699	1.2231	115	90.9735	1.6935	5.1093
52	32.0593	0.0739	0.1152	116	95.2863	3.1051	10.0279
53	30.0398	0.0696	0.1162	117	88.1141	6.1320	22.8394
54	28.4575	0.0706	0.1392	118	98.2369	14.2494	53.2881
55	122.9108	117.8151	422.9925	119	106.1378	1.7406	5.3467
56	75.7503	9.9113	20.6392	120	105.9322	6.2828	20.1205

Table A.1 Continued

57	103.6334	17.0152	44.5223	121	101.7614	122.6593	473.8919
58	42.6930	2.6194	8.0912	122	104.4759	0.1017	0.1653
59	37.0388	5.5358	20.2768	123	101.2885	0.1077	0.1257
60	44.9534	6.0386	22.4401	124	91.4579	0.4946	1.2872
61	28.4795	0.0953	0.2297	125	82.3724	2.8142	9.6330
62	25.8045	0.0829	0.2068	126	98.1856	45.0803	172.3485
63	124.3925	248.3275	922.4570	127	113.0229	0.5568	0.8065
64	125.3803	565.0660	2139.0227	128	129.6634	1.9054	5.1472

A.2 Analysis results of Forward model

All the root mean squared error, new error measure, and its standard deviation of each channel between the true EEG signals and the estimated EEG signals from Forward model are calculated and shown in Table A.2. The root mean squared error ranges from 2.0008 to 2.7261; while the new error measure ranges from 0.7981 to 1.5306.

Table A.2: Results of Forward model performance evaluation in three different parameters: root mean square of error (RMS), customize error (Error), and the standard deviation of Error (std.)

EEG channel	RMS	Error	std.
F_{p1}	2.5419	0.8764	4.9142
F_7	2.5362	1.0448	4.1353
F_3	2.3860	0.9628	3.2328
F_z	2.7261	1.5306	6.0478
T_3	2.2092	1.2591	5.0811
C_3	2.3368	0.7981	2.4571
T_5	2.1505	1.0541	3.1589
P_3	2.5236	1.1482	2.9866
P_z	2.0987	0.7988	3.1662
O_1	2.0008	1.0134	2.9818

A.3 Analysis results of PC estimation

All the root mean squared error, new error measure, and its standard deviation of each channel between the true principal components from NLPCA model and the estimated principal components from Inverse estimation are calculated and shown in Table A.3. The root mean squared error ranges from 0.3169 to 1.8808; while the new error measure ranges from 0.0388 to 1.1516.

Table A.3: Results of inverse model performance evaluation from the perspective of principal components in three different parameters: root mean square of error (RMS), customize error (Error), and the standard deviation of Error (std.). PC_i represents the i^{th} principal component.

EEG ch.	P.C.	RMS	Error	std.	EEG ch.	P.C.	RMS	Error	std.
F_{p1}	PC_1	1.5147	0.4380	2.1368	C_3	PC_1	0.7778	0.3504	0.5943
	PC_2	1.2078	0.4944	1.1568		PC_2	0.8182	0.2284	0.6895
	PC_3	1.5364	0.4865	2.4201		PC_3	0.6077	0.1012	0.2585
	PC_4	0.7978	0.2232	0.8904		PC_4	0.7438	0.3242	0.5292
	PC_5	0.9896	0.4029	1.0794		PC_5	0.5583	0.1558	0.3389
F_7	PC_1	0.7338	0.1575	0.6650	T_5	PC_1	0.9714	0.2454	0.6418
	PC_2	0.7613	0.1410	0.4811		PC_2	1.0154	0.4684	1.0988
	PC_3	0.6589	0.2861	0.4656		PC_3	1.0563	0.6134	1.1704
	PC_4	0.5376	0.0963	0.3105		PC_4	0.6773	0.2171	0.4073
	PC_5	0.4761	0.1292	0.2177		PC_5	0.9814	0.3638	0.8401
F_3	PC_1	0.8979	0.1441	0.5439	P_3	PC_1	0.6851	0.2338	0.5243
	PC_2	1.0931	0.1756	0.8160		PC_2	0.5565	0.1033	0.2011
	PC_3	0.9164	0.2113	0.5607		PC_3	0.3292	0.0454	0.0844
	PC_4	0.8941	0.2166	0.6445		PC_4	0.6932	0.2301	0.4574
	PC_5	0.7431	0.2587	0.4685		PC_5	0.3169	0.0388	0.0883
	PC_1	0.3963	0.0842	0.1601		PC_1	0.9861	0.1626	0.4348

Table A.3 Continued

F_z	PC_2	0.4649	0.1322	0.2574	P_z	PC_2	1.2928	0.4445	1.1546
	PC_3	0.6998	0.3253	0.6241		PC_3	1.3147	0.4606	1.1304
	PC_4	0.5820	0.1829	0.3458		PC_4	1.1893	0.4782	1.2741
	PC_5	0.4995	0.0545	0.1741		PC_5	1.0919	0.5384	1.1375
T_3	PC_1	0.7792	0.3449	0.7324	O_1	PC_1	1.8808	1.1516	2.8391
	PC_2	0.9000	0.3299	0.7714		PC_2	1.6910	0.5262	1.6660
	PC_3	1.2756	0.3683	1.2611		PC_3	1.4110	0.4394	1.2558
	PC_4	0.7352	0.2779	0,7213		PC_4	1.4826	0.6160	1.5474
	PC_5	0.8823	0.3412	0.7258		PC_5	1.6785	0.4176	1.7389

A.4 Analysis results of Inverse solution

All the root mean squared error, new error measure, and its standard deviation of each channel between the true ECoG signals and the reconstructed ECoG signals from the Inverse solution are calculated and shown in Table A.4. The root mean squared error ranges from 48.6886 to 414.7277; while the new error measure ranges from 18.1474 to 1735.6951. Note again that there is one channel: 68 with the value N/A since we discard this channel due to its location which is not in the effective cover range of EEG channels.

Table A.4: Results of Inverse model performance evaluation in three different parameters: root mean square of error (RMS), customize error (Error), and the standard deviation of Error (std.)

ECoG	RMS	Error	std.	ECoG	RMS	Error	std.
1	82.5782	353.8335	2043.4379	65	92.8788	121.2411	1291.9756
2	124.6050	199.1545	1966.6904	66	79.7336	107.0795	1203.8006
3	56.1094	375.8717	1416.3736	67	111.9855	205.5284	2521.2710
4	149.7877	126.5934	1646.4008	68	N/A	N/A	N/A
5	79.9216	219.5283	1220.1769	69	156.6719	483.2806	3798.3744
6	156.6867	161.0342	1907.9436	70	149.8655	911.4109	5418.7921
7	97.6252	206.1679	1392.8955	71	173.3398	1735.6951	9390.8913
8	57.1343	199.5062	1072.9339	72	113.3875	55.3532	1161.9205
9	189.6361	494.8700	5541.7980	73	98.2072	33.1117	947.9128
10	158.8947	105.4677	1840.4390	74	83.3926	62.8879	767.5892
11	99.1535	480.9051	3520.2283	75	94.8511	58.7000	1092.2751
12	112.3630	401.1267	3192.1120	76	189.7175	120.1975	1489.0033
13	131.9652	470.7259	4598.3439	77	138.4153	90.0974	1484.0560
14	83.1836	232.2012	1519.0478	78	98.2596	91.7158	1428.6460

Table A.4 Continued

15	56.4545	315.6940	1162.3018	79	127.7891	67.5061	733.9116
16	48.6886	322.7459	1382.3783	80	103.6208	63.0105	890.0944
17	109.7627	378.4752	2530.6771	81	179.7379	98.5191	1556.8589
18	99.4724	346.2185	2269.8189	82	168.0090	74.9528	1533.6865
19	188.2853	174.0303	2026.7644	83	84.0108	27.9707	392.1036
20	135.0025	122.5395	1504.0662	84	104.7791	18.1474	387.9468
21	140.2051	206.5632	3020.9121	85	109.2491	53.7643	1410.5839
22	120.2323	35.0488	775.4545	86	131.5736	45.8236	1208.2285
23	116.1898	47.7376	544.7886	87	137.4679	35.1103	621.6119
24	107.0393	49.8165	989.6558	88	110.5568	22.1880	390.8048
25	125.3667	148.3142	2938.6055	89	103.6629	29.2111	832.7075
26	162.2698	201.6459	2162.2621	90	135.0319	71.4883	1146.0806
27	120.9522	107.3104	1264.6560	91	203.6224	57.4467	1562.8416
28	90.3072	70.4999	729.0816	92	140.4131	40.6028	808.3251
29	103.3431	108.0399	1223.6339	93	414.7277	238.9737	5802.7172
30	103.8499	97.2288	1123.3351	94	225.0121	157.5457	3147.7180
31	74.6533	53.6674	802.4888	95	115.1397	138.2185	1541.4357
32	70.6985	37.1007	527.4563	96	136.2489	156.1328	1515.4958
33	80.4936	38.7871	418.9641	97	136.9904	154.3238	1955.9910
34	99.0800	43.4966	551.7416	98	141.4000	245.4511	3026.8820
35	99.8234	71.7364	1312.6637	99	142.0088	233.1020	3058.6688
36	101.6203	94.0397	1327.4299	100	172.7274	74.6445	1297.2949
37	140.5184	114.9154	1608.7000	101	207.4617	234.4015	3574.5805
38	108.5411	55.1195	1358.9641	102	177.6105	55.3676	945.8527
39	134.5745	11.5059	289.0109	103	159.2966	82.1714	1322.1910

Table A.4 Continued

40	107.6116	55.7188	725.9240	104	212.8247	33.0839	844.1385
41	59.1463	36.6228	508.4256	105	226.7145	29.9291	585.2207
42	56.0714	52.0190	474.9992	106	216.4831	74.5083	1805.7478
43	84.3389	64.1742	646.6743	107	134.9267	178.1672	2333.0829
44	107.2035	94.2668	1320.9039	108	136.1898	153.4231	2099.6096
45	104.8479	20.2801	343.2886	109	156.9559	103.3209	2134.1609
46	134.6178	93.6301	2294.8181	110	170.6734	145.1031	1871.0827
47	130.2157	149.4754	1998.4482	111	176.3483	210.3005	3528.3137
48	128.9560	148.1413	2142.6898	112	181.1403	152.1874	1650.1761
49	127.1393	57.3676	1062.4563	113	158.3583	186.9760	2517.0193
50	124.2501	41.7925	747.6893	114	214.9799	204.6184	2853.1234
51	112.3561	85.7917	959.3270	115	197.3511	218.1982	3276.2471
52	85.2393	29.6205	406.6752	116	186.7538	161.9592	3602.3267
53	64.7339	24.3797	493.1918	117	166.5832	110.8369	1490.8062
54	74.7883	80.1776	799.6879	118	130.8762	201.7432	3384.2935
55	116.9870	109.1759	1853.4600	119	153.4429	58.5686	1067.0082
56	109.2837	117.7193	1359.4090	120	175.6337	51.8143	2376.4312
57	108.4085	139.6940	1416.9606	121	205.8283	148.3425	4487.6894
58	115.8459	59.7578	672.8922	122	157.2780	132.2100	1316.9612
59	88.5289	60.9672	1319.8835	123	186.1616	194.5157	2911.5010
60	94.9522	99.5207	1009.8628	124	194.8465	193.6543	4068.1972
61	66.9741	25.6114	376.8541	125	181.4183	208.3093	3620.2635
62	50.6700	22.6938	238.1428	126	158.3479	176.0345	2271.2336
63	210.7821	361.9639	4100.1919	127	147.9896	178.9813	2900.4999
64	84.6005	82.8540	1021.5202	128	160.8484	172.1773	2302.0551

Bibliography

[1] Abdulkader, S. N., Atia, A., & Mostafa, M. S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2), 213-230.

[2] Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., ... & Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398), 372.

[3] Grant, B. (2014, June). Paralyzed Man Kicks Off World Cup. *The Scientist*. Retrieved from https://www.the-scientist.com/

[4] Leuthardt, E. C., Roland, J. L., & Ray, W. Z. (2014, November). Neuroprosthetics. *The Scientist*. Retrieved from https://www.the-scientist.com/

[5] Niedermeyer, E., & da Silva, F. L. (Eds.). (2005). Electroencephalography: basic principles, clinical applications, and related fields. *Lippincott Williams & Wilkins*.

[6] Hmlinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalographytheory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2), 413.

[7] Lebedev, M. A., & Nicolelis, M. A. (2017). Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation. *Physiological reviews*, 97(2), 767-837.

[8] Pascual-Marqui, R. D. (1999). Review of methods for solving the EEG inverse problem. *International journal of bioelectromagnetism*, 1(1), 75-86.

[9] Grech, R., Cassar, T., Muscat, J., Camilleri, K. P., Fabri, S. G., Zervakis, M., ... & Vanrumste, B. (2008). Review on solving the inverse problem in EEG source analysis. *Journal of neuroengineering and rehabilitation*, 5(1), 25.

[10] Lodish, H., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D., & Darnell, J. (2000). Neurotransmitters, synapses, and impulse transmission.

[11] Kybic, J., Clerc, M., Abboud, T., Faugeras, O., Keriven, R., & Papadopoulo, T. (2005). A common formalism for the integral formulations of the forward EEG problem. *IEEE transactions on medical imaging*, 24(1), 12-28.

[12] Marin, G., Guerin, C., Baillet, S., Garnero, L., & Meunier, G. (1998). Influence of skull anisotropy for the forward and inverse problem in EEG: simulation studies using FEM on realistic head models. *Human brain mapping*, 6(4), 250-269.

[13] Von Ellenrieder, N., Muravchik, C. H., & Nehorai, A. (2005). A meshless method for solving the EEG forward problem. *IEEE Transactions on Biomedical Engineering*, 52(2), 249-257.

[14] Ding, L., Lai, Y., & He, B. (2004). Low resolution brain electromagnetic tomography in a realistic geometry head model: a simulation study. *Physics in Medicine & Biology*, 50(1), 45.

[15] Baillet, S., Mosher, J. C., & Leahy, R. M. (2001). Electromagnetic brain mapping. *IEEE Signal processing magazine*, 18(6), 14-30.

[16] Xu, X. L., Xu, B., & He, B. (2004). An alternative subspace approach to EEG dipole source localization. *Physics in Medicine & Biology*, 49(2), 327.

[17] Hallez, H., Vanrumste, B., Grech, R., Muscat, J., De Clercq, W., Vergult, A., ... & Lemahieu, I. (2007). Review on solving the forward problem in EEG source analysis. *Journal of neuroengineering and rehabilitation*, 4(1), 46.

[18] Berthold, M. R., Cebron N., Dill F., Gabriel T. R., Kötter, T., Meinl, T., Ohl P., Sieb, C., Thiel, K., & Wiswedel B. (2007). KNIME: The Konstanz Information Miner In Studies in Classification, Data Analysis, and Knowledge Organization. *Springer*

[19] Bartholomew, D. J., Steele, F., Galbraith, J., & Moustaki, I. (2008). Analysis of multivariate social science data. *Chapman and Hall/CRC*.

[20] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6), 417.

[21] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2), 233-243.

[22] Caudill, M. (1990). AI expert: Neural network primer. *Miller Freeman Publications 1990*.

[23] Deb, S., & Dixit, U. S. (2008). Intelligent machining: computational methods and optimization. *Machining* (pp. 329-358). Springer, London.

[24] Sjlie, D. (2011). Reality-based brain-computer interaction (Doctoral dissertation, Department of Computing Science, Ume University).

[25] Montbrun-Di Filippo, J., Delgado, M., Brie, C., & Paynter, H. M. (1991). A survey of bond graphs: theory, applications and programs. *Journal of the Franklin Institute*, 328(5-6), 565-606.

[26] Fernandez, B., Prabhudesai, A. V., Murty, V. V., Gupta, R., & Chang, W. R. (1992). Neurobondgraphs: modeling environment of nonlinear dynamic systems using neural networks and bond graphs. In *Neural Networks in Manufacturing and Robotics* (Vol. 57, pp. 75-90). Publ by ASME.

[27] Choudhary, I., Assaleh, K., & AlHamaydeh, M. (2012, April). Nonlinear AutoRegressive eXogenous Artificial Neural Networks for predicting buckling restrained braces force. In *Mechatronics and its Applications (ISMA), 2012 8th International Symposium on* (pp. 1-5). IEEE.

[28] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

[29] Olah, C. (2015, August 27). Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[30] Sain, M., & Massey, J. (1969). Invertibility of linear time-invariant dynamical systems. *IEEE Transactions on automatic control*, 14(2), 141-149.

[31] Darouach, M., Zasadzinski, M., & Xu, S. J. (1994). Full-order observers for linear systems with unknown inputs. *IEEE transactions on automatic control*, 39(3), 606-609.

[32] Masoudnia, S., & Ebrahimpour, R. (2014). Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2), 275-293.

[33] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1), 79-87.

[34] LeCun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal brain damage. In *Advances in neural information processing systems* (pp. 598-605).

[35] Hassibi, B., Stork, D. G., & Wolff, G. J. (1993). Optimal brain surgeon and general network pruning. In Neural Networks, 1993., *IEEE International Conference* on (pp. 293-299). IEEE.

[36] Nagasaka, Y., Shimoda, K., & Fujii, N. (2011). Multidimensional recording (MDR) and data sharing: an ecological open research and educational platform for neuroscience. *PloS one*, 6(7), e22561.

[37] Luck, S. J. (2014). An introduction to the event-related potential technique. *MIT press*.

[38] Yunhui, S., & Qiuqi, R. (2004). Continuous wavelet transforms. In *Signal Processing*, 2004. Proceedings. ICSP'04. 2004 7th International Conference on (Vol. 1, pp. 207-210). IEEE.

Vita

Yin-Jui Chang was born in Yunlin, Taiwan. He attended Taipei Municipal Song Shan Senior High School from 2009 to 2011. Then he entered National Taiwan University, where he received Bachelor of Science in Mechanical Engineering in 2015. Following his graduation, he continued to study at the University of Texas at Austin in Mechanical Engineering. While he was a graduate student, he served as a graduate assistant and teaching assistant. His interests in mechanical engineering include machine learning, Brain-Computer Interface, rehabilitation, and the modeling and control of dynamic systems.

Address: derek706586@utexas.edu

This thesis was typeset with LaTeX 2$_\varepsilon$[1] by the author.

[1] LaTeX 2$_\varepsilon$ is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.